UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

SecNav

Autor: Roger Raventós Calle

Director: Raúl Roca Cánovas
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 13 de setembre de 2020

Abstract

SecNav is an extension for Google Chrome that looks for improve your feelings and security while navigating through the network using local and remote data thanks to a little example server. The extension using these data can decide if the browser can connect, download or open new tabs or windows depending of the domain, page, scheme or other filters provided by the user or the server. This project explains the design, implementation, working, uses of this extension, the connection with the server and the extra utilities that the server contributes with specific examples.

Resum

SecNav és una extensió per Google Chrome que busca millorar l' experiència i seguretat de navegació web. Per aconseguir-ho maneja dades de forma local i remota gràcies a un petit servidor d'exemple. Aquestes dades permetran a l'extensió decidir si el navegador es podrà connectar, descàrregar fitxers o obrir altres pestanyes o finestres d'un domini, pàgina en concret o inclús un protocol com 'http', 'https' o 'ftp'. A més disposa d'altres opcions de filtres donats per l'usuari o el servidor. Aquest treball explica amb detall el disseny, la implementació, funcionament, casos d'ús d'aquesta extensió, la connexió d'ella amb el servidor i les utilitats extra que pot arribar a aportar el servidor amb casos concrets d'exemple.

Resumen

SecNav es una extensión para Google Chrome que se centra en mejorar la experiéncia y seguridad de navegación web. Para consegirlo gestiona datos de manera local y remota gracias a un pequeño servidor de ejemplo. Todos estos datos permitirán a la extensión decidir si el navegador se podrá conectar, descargar archivos o abrir ventanas emergentes de un dominio, página o incluso de un protocolo como 'http', 'https', o incluso 'ftp'. Además dispone de de otras opciones de filtros otorgadas por los mismos usuarios o el servidor remoto. Este trabajo explica en detalle el diseño, implementación, utilización, casos de uso, la conexión entre ella y el servidor y las diferentes utilidadades extra que el servidor puede llegar a aportar a la extensión.

Agraïments

Vull agrair a primer de tot a la meva família, després als meus amics i al meu tutor per tenir, tots, la paciència que han tingut amb mi.

Índex

1 Introducció

Context

Navegar per Internet és una de les accions més habituals del dia a dia de milions de persones, encara que sense saber-ho o inclús sabent-ho una de les més perilloses. Hi ha un munt de pàgines malicioses que volen apoderar-se del control dels recursos del teu ordinador, altres et volen enganyar per extreure't les dades personals i treure'n un profit econòmic, es podrien estar hores enumerant els perills de la web. És cert que hi ha Firewalls i altres recursos que es poden utilitzar per protegir-se, però també és cert que mai està de més tenir una capa extra de seguretat, en aquest cas de cost molt reduït a nivell de hardware i més accessible.

En aquest treball es construirà una extensió des de zero, SecNav, que ens afegirà una capa més de seguretat en la navegació, la qual podrà gestionar un servidor per nodrir-la amb dades i així fer l'experiència encara més segura i còmode. Aquesta experiència també és gràcies a les dades locals que pot aportar l'usuari amb un parell o tres de clics, ja sigui amb importació de configuració que conté pàgines que es consideren perilloses o segures depenent de com es vulgui fer funcionar l'extensió amb llistes negres o blanques.

Motivació

Sempre m'havia cridat l'atenció el tema de les extensions i aplicacions de Chrome, hi ha de tot tipus i es poden fer moltíssimes coses que ni tans sols sé. D'altra banda la ciberseguretat em pensava que era una branca de la informàtica, però he descobert durant aquest any que ha de ser un dels troncs principals de l'arbre sense haver-ne profunditzat massa.

És per això que ajuntar dos temes que em semblen curiosos és una oportunitat per descobrir poc a poc aspectes i punts de vista dels dos mons. A part que a mi m'encanta programar coses noves i fer que vagin el millor possible, ara he descobert que també ha de ser el menys insegur possible.

1.1 Objectius

1.1.1 Objectiu principal

L'objectiu principal d'aquest treball és realitzar el disseny, la implementació i exemples de casos d'ús de SecNav, una extensió de Chrome per fer més segura i còmode la navegació per la xarxa.

1.1.2 Objectius específics

Per a realitzar l'objectiu principal s'han de complir els següents objectius específics:

- **Disseny de l'extensió de Chrome:** Per a realitzar el disseny primer s'ha de realitzar un petit estudi de la plataforma a utilitzar, en aquest cas una extensió de Chrome i a partir de la informació obtinguda realitzar el disseny de l'extensió amb aquestes característiques:
  - Gestió de descàrregues
  - Gestió de pàgines
  - Gestió d'emergents
  - Gestió d'elements
  - Gestió de permisos
- **Implementació de l'extensió:** S'ha d'implementar el disseny de l'extensió i juntament crear un petit servidor d'exemple per nodrir l'extensió i poder comprovar la implementació de la mateixa.

1.2 Planificació

A grans trets es divideix en dos grans blocs i un tercer més reduït:

El primer, però no per això el menys important és el del disseny de l'extensió, ja que per a després realitzar una implementació sense necessitats de grans modificacions del disseny original, d'aquesta manera se li dedicarà tres setmanes, ja que també s'ha de decidir entre eines i llibreries auxiliars tant per la part visual com la oculta i dissenyar la part visual.

El segon gran bloc és el de la implementació del disseny de l'extensió, el qual durarà dos mesos, ja que és un projecte amb tasques molt específiques però fa ús de diferents dades que hauran de ser accessibles a la modificació en múltiples escenaris, cosa que pot provocar un canvi en el disseny original i així tenir marge de temps en aquest aspecte.

Per acabar tenim un tercer bloc, aquest molt més petit que consisteix en fer diferents demostracions el qual tindrà una duració d'una setmana, ja que és configurar l'extensió, el servidor i el sistema operatiu amb diferents configuracions i veure com es comporta.

1.2.1 Taules i Diagrames

Tasques	Data inici	Duració	Data final
Estudi de desenvolupament d'una extensió de chrome	juny	5	10-jun
Elecció de tema de disseny i utilitats	juny	5	15-jun
Disseny de l'estructura de l'extensió	juny	10	25-jun
Planificació de la implementació	juny	5	30-jun
Implementació de les utilitats	juliol	15	15-jul
Implementació de les diferents seccions	juliol	30	16-ago
Implementació i petit disseny del servidor	agost	7	23-ago
Proves d'ús	agost	7	30-ago

Figura 1: Planificació inicial del projecte

Tasques	Data inici	Duració	Data final
Estudi de desenvolupament d'una extensió de chrome	juliol	5	05-jul
Elecció de tema de disseny i utilitats	juliol	5	10-jul
Disseny de l'estructura de l'extensió	juliol	7	17-jul
Planificació de la implementació	juliol	3	20-jul
Implementació de les utilitats	juliol	15	05-ago
Implementació de les diferents seccions	agost	25	31-ago
Implementació i petit disseny del servidor	setembre	4	04-sep
Proves d'ús	setembre	6	10-sep

Figura 2: Realització final del projecte

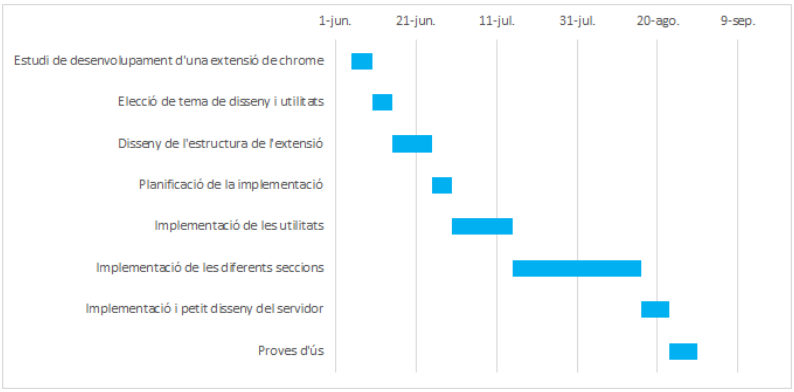


Figura 3: Diagrama de Gantt inicial del projecte

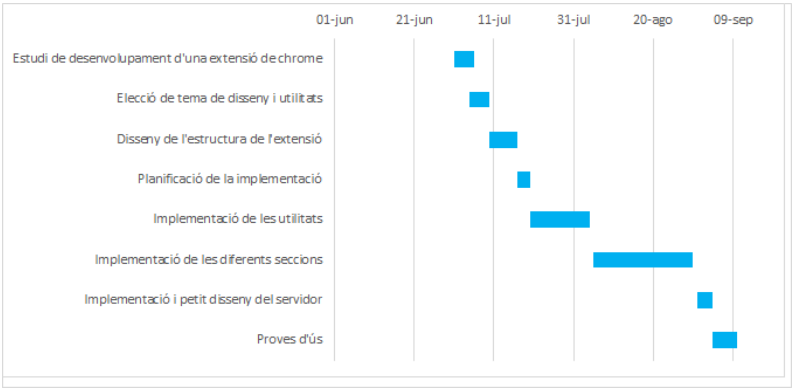


Figura 4: Diagrama de Gantt final del projecte

Estructura de la Memòria

L'estructura de la memòria es desglossa en els següents capítols:

- **Introducció:** El seu objectiu és definir els objectius del projecte i com s'ha plantejat.
- **Anàlisi:** El seu objectiu és entendre la plataforma que s'utilitzarà al projecte.
- **Disseny:** Es descriu el disseny inicial optat i les raons d'aquest disseny.
- **Implementació:** Es descriu la implementació del disseny original i les modificacions que s'han realitzat durant la mateixa.
- **Exemples d'ús:** Es posen exemples de la utilització de l'extensió amb usos de les diferents funcionalitats.
- **Conclusions i treball futur:** Es detallen les conclusions i possibles pròximes iteracions del treball.
- **Annex:** S'inclou una guia per a l'execució dels diferents elements del projecte i en diferents sistemes operatius.

2 Anàlisi de la plataforma de treball

Abans d'avançar amb el disseny i posteriorment la implementació, és necessari analitzar el problema i com abordar-lo per poder aconseguir complir els objectius establerts.

És necessari que l'extensió sigui distribuïda, escalable i que es pugui executar sense que l'usuari ho noti, excepte quan sigui estrictament necessari o ell vulgui intervenir-hi.

També és necessari un registre persistent on es vagin guardant les dades i errors per tal d'analitzar-les posteriorment per tal de solucionar errors, o millorar la configuració inicial o simplement fer un estudi extern.

2.1 Introducció

Ja que es vol crear una extensió, és important saber què és, quines són les diferents opcions d'execució i a quins recursos interns es requerirà l'accés i com explotar-los.

Una extensió de Google Chrome és un petit software que et permet personalitzar la teva experiència utilitzant el navegador. Per fer-ho utilitza tecnologies com són HTML, css i Java-Script. Una extensió encara que realitzi moltes funcionalitats i disposi de múltiples components diferents ha d'estar enfocada a un únic propòsit.[2]

2.2 Manifest.json

Aquest fitxer JSON és obligatori per a qualsevol extensió, és el fitxer que enllaça tots els recursos de l'extensió perquè pugui executar-se, qualsevol error al manifest farà que l'extensió no arribi ni a habilitar-se per ser executada.

El Manifest.json suporta fins a 60 camps de configuració principals, dels quals 3 són obligatoris, 3 recomanats i la resta opcionals.

Els camps obligatoris són els següents:

- ***manifest_version*:** Defineix la versió del manifest.
- ***name*:** Permet escollir quin serà el nom que rebrà l'extensió.
- ***version*:** Permet escollir el nom de la versió actual de l'extensió.

Google en la seva documentació oficial recomana els següents camps:

- ***default_locale*:** Si l'extensió utilitza 'locales' per internacionalitzar-se s'ha de definir quin és el llenguatge que haurà d'utilitzar l'extensió en cas de no haver implementat el llenguatge amb el que treballa el navegador.
- ***description*:** Permet descriure breument l'extensió.
- ***icons*:** Permet definir un conjunt d'icones amb diferents mides.

Pel que fa als camps opcionals es centrarà en els més importants pel nostre projecte que, entre d'altres, són:

- ***permissions*:** Permet concedir-li permisos a l'extensió per a utilitzar APIs de les APIs de Chrome, com controlar pestanyes, descàrregues, sol·licituds entre d'altres.
- ***background*:** Permet tenir codi en execució constantment, on es podrà executar la lògica del programa.
- ***options_page*:** Permet tenir una pàgina HTML de configuració per l'extensió.
- ***content-scripts*:** Permet injectar codi a les pàgines predefinides mantenint la comunicació amb el background de l'extensió.
- ***browser_action*:** Permet tenir un petit pop-up i mostrar notificacions a l'icona de l'extensió.
- ***update_url*:** Permet a l'extensió buscar actualitzacions en aquesta direcció, cal mencionar que si l'extensió no forma part de la botiga de Google aquesta url només serà utilitzada en ordinadors gestionats on es permeti l'ús d'extensions externes i en dispositius Linux.

2.3 Internacionalització

Aquesta funcionalitat mencionada en el manifest permet a l'extensió permet a una extensió internacionalitzar-se d'una forma nativa, senzilla i eficient utilitzant la infraestructura `chrome.i18n`.

Per utilitzar-la s'ha de crear un fitxer *messages.json* i guardar-lo dins del directori *_locales/codi_local* sent *codi_local* un dels codis locals suportats per Google.(Figura 5)

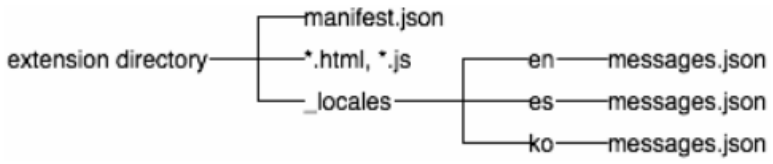


Figura 5: Diagrama del funcionament del *locale*

2.4 APIs de Chrome

Chrome disposa de moltes API[1] que poden utilitzar les seves extensions, sempre i quan es tinguin els permisos necessaris registrats correctament al manifest. S'organitzen en APIs estables, en versió beta, de desenvolupadors i en experimentals.

Pel que són les necessitats de l'extensió només es farà èmfasis en les següents APIs estables:

- ***browserAction*:** Permet obtenir i modificar el *browserAction* sempre i quan s'hagi registrat al manifest.
- ***cookies*:** Permet obtenir i modificar *cookies* sempre i quan s'hagi registrat al manifest el permís necessari i addicionalment també s'ha de registrar el permís de a quines direccions URL serà capaç d'accedir a les *cookies*.
- ***donwloads*:** Permet tenir un control total de les descàrregues, des de cancel·lar-ne una en curs com esborrar un fitxer ja descarregat passant per gestionar l'historial de descàrregues, necessita tenir registrat el permís al manifest.
- ***i18n*:** Gestiona la Internacionalització mencionada específicament a l'apartat anterior.
- ***runtime*:** Permet comunicar-se amb el background, escoltar i respondre a esdeveniments, obtenir informació del manifest, bàsicament utilitats generals per l'extensió que no requereix cap tipus de permís addicional al manifest.
- ***storage*:** Permet emmagatzemar dades tant en local com en 'sync', perquè estiguin disponibles a tots els teus dispositius vinculats i compatibles amb l'extensió, per fer-ho utilitza un format de dades de diccionari amb una clau i un valor. Necessita especificar el permís al manifest.
- ***tabs*:** Permet gestionar les pestanyes obertes o crear-ne de noves, no requereix permisos a no ser que es vulgui accedir a atributs com la URL, que en aquest cas s'haurà d'especificar al manifest.
- ***webRequest*:** Permet gestionar les sol·licituds, des d'observar passant per modificar o inclús bloquejar-les, requereix de permís registrat al manifest, permís addicional dels dominis en què es podrà tenir accés a les sol·licituds i addicionalment permís per bloquejar sol·licituds si així s'escau.

2.5 Polítiques de Chrome per a ordinadors gestionats

Google disposa d'un munt de polítiques per configurar el seu navegador en ordinadors gestionats. Els ordinadors gestionats són aquells que, en la majoria dels casos, s'utilitzen en empreses o organitzacions i tenen associats unes regles que venen imposades per un usuari de major rang.

En aquest treball no s'utilitzaran totes les polítiques però s'enumeraran les que poden afectar directament i indirectament a l'extensió:

- ***ExtensionSettings*:** Aquesta política permet gestionar les extensions, des de bloquejar-ne la instal·lació fins obligar a la instal·lació passant per un munt de paràmetres molt específics que no són tant rellevants en aquest treball.

- ***DeveloperToolsAvailability*:** La política gestiona la disponibilitat del panell de desenvolupador.
- ***BrowserGuestModeEnabled*:** Aquesta altra política permet habilitar o deshabilitar l'usuari d'invitat dins del navegador.
- ***IncognitoModeAvailability*:** La última política permet gestionar la utilització del mode incògnit.

Hi ha altres polítiques que es podien haver mencionat, com *ExtensionInstallForcelist* o *BlockExternalExtensions* però s'han obviat ja que es podria dir que són parts d'*ExtensionSettings* i no s'ha considerat necessari.

3 Disseny

El disseny de l'extensió es pot classificar en quatre seccions diferents, elements de disseny, el disseny del background, el disseny de configuració i el disseny del PopUp:

3.1 Elements de disseny

L'extensió disposarà d'una part amb relació directa amb l'usuari, per tant, és important seguir uns mateixos patrons al llarg del disseny visual i concretar, en aquest cas, el framework visual utilitzat.

3.1.1 Logo de l'extensió

El disseny del logo de l'extensió utilitza vàries metàfores visuals per transmetre el seu missatge. Es pot observar el que és la base d'un vaixell sobre l'aigua fent referència en aquest cas a la navegació per Internet. El cadenat que es situa al lloc de les vel·les fa referència a la seguretat. Al ajuntar les dues parts es vol donar a entendre el concepte de navegació segura. (Figura 6)

Figura 6: Logo de l'extensió

3.1.2 MDBootstrap

Per a realitzar la part visual de l'extensió s'ha decidit utilitzar Material Design for Bootstrap[7] a partir d'ara en endavant MDB, concretament la versió que utilitza jquery, ja que ja s'havia decidit utilitzar jquery com a eina per a realitzar accions al background, és molt potent, senzilla d'utilitzar i no suposa una gran càrrega pel sistema. MDB proporciona uns acabats agradables a la vista i molt minimalistes, exactament el que es vol buscar en una extensió.

MDB és un framework que utilitza com a base bootstrap i que està modificat per un estil més propi i més opcions tant de css com de Java-Script, té dues versions. La gratuïta amb els components més bàsics i la de pagament anomenada PRO que ofereix un munt més de opcions de personalització per a qualsevol projecte. En aquest treball s'ha optat per la versió gratuïta ja que és més que suficient per a les necessitats que comporta.

Figura 7: Paleta de colors

La paleta de colors que s'utilitzarà al llarg del treball pels diferents elements és la predeterminada del paquet MDB ja que s'ha considerat que eren adequats al ser colors que la gent està acostumada a veure i interpretar el seu segon significat en molts casos. Per exemple un semàfor utilitza el vermell com a prohibició, el taronja com a advertiment i el verd com a èxit. (Figura 7)

3.2 Background

El disseny del background s'ha pensat com si fos el backend d'un servidor o d'una aplicació, és a dir que la lògica principal i la comunicació amb les diferents fonts de dades es realitza a través seu.

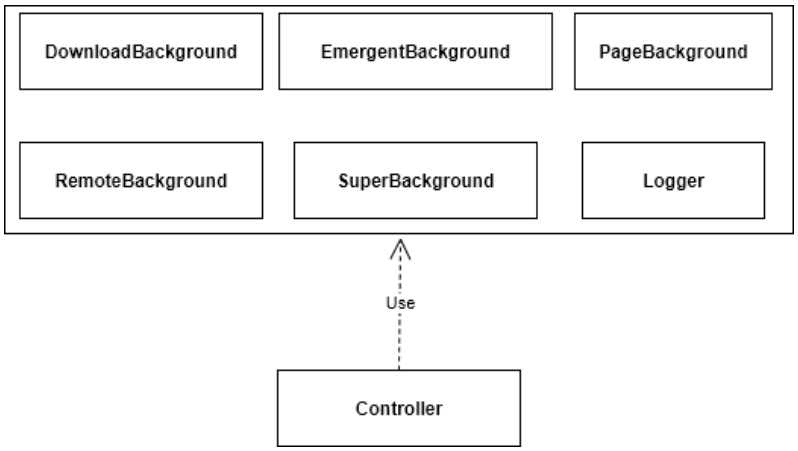


Figura 8: Diagrama de classes principals

La figura 8 és un diagrama de classes principals de com s'organitza el background. Existeix la presència d'un controlador Controller que s'encarrega d'inicialitzar i controlar totes les interaccions de la resta de l'extensió amb totes les classes singleton que utilitza. Un singleton[11] és un patró de disseny que es fa servir per evitar que una classe tingui més d'una instància. En aquest cas s'ha utilitzat aquest recurs

ja que encara que es tinguin 20 pestanyes obertes les accions seran sempre seran realitzades sobre les mateixes dades, per tant en aquest cas no té cap sentit tenir dues instàncies d'una mateixa classe.

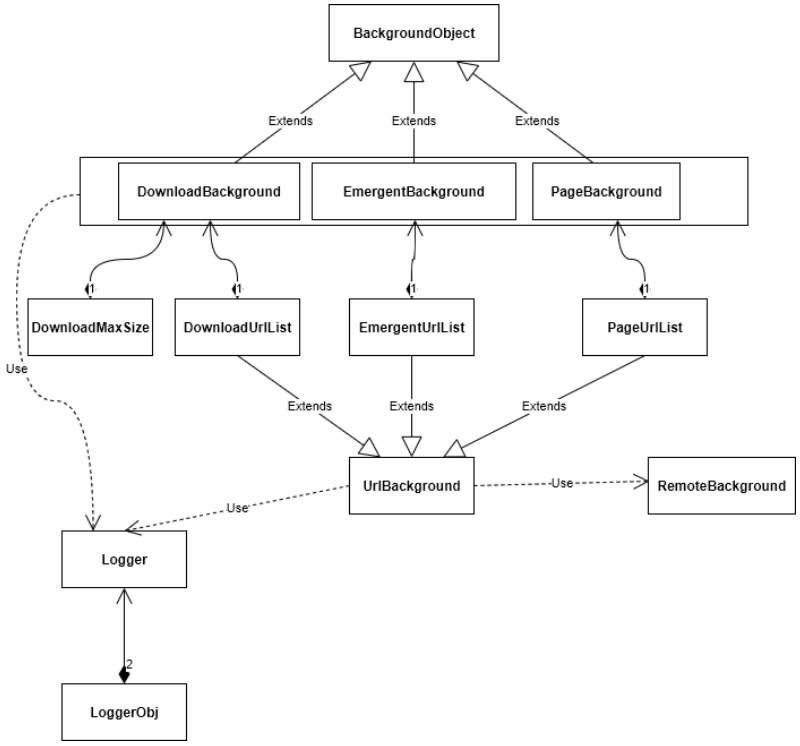


Figura 9: Diagrama de classes dels elements del background

Tant `Downloadbackground` com `Emergentbackground` com `Pagebackground` hereden de `backgroundObject`, ja que estem treballant sobre Java-Script no es pot crear una classe abstracta de forma eficaç perquè els fills implementin les funcions no implementades per la classe mare, de tota manera les tres comparteixen certs atributs i funcions derivats d'aquests.

La classe `Urlbackground` utilitzarà a `Remotebackground`, que s'encarrega de les operacions amb el servidor, sol·licitar dades al servidor i gestionar la connexió, aquestes dades seran utilitzades pels fills d'aquesta, `DownloadUrlList`, `EmergentUrlList` i `PageUrlList` per determinar si per la seva part és necessari realitzar una acció sobre l'esdeveniment que es produeixi.

Tots els errors i passos realitzats que tinguin certa importància s'enviaran a la classe Logger que s'encarregarà de guardar el registre al log que es mostrarà a l'usuari final o al log de desenvolupament per ajudar a trobar i solucionar errors que es puguin produir durant l'execució del programa. (Figura 9)

Figura 10: Diagrama de classes i fitxers independents

La figura 10 és un diagrama on es mostren tres classes estàtiques que podran ser utilitzades per tots els elements del background si així fos necessari, aquestes classes estan pensades per accions concretes i generals. En el cas de la classe *Import* requerirà d'una implementació en cada una de les classes que gestionin dades importables.

Aquest diagrama també ens mostra quatre fitxers de Java-Script que a part de poder ser utilitzats per tot el background també podrà ser utilitzats per la lògica de la part visual de l'extensió i així mantenir el mateix criteri d'una forma més senzilla.

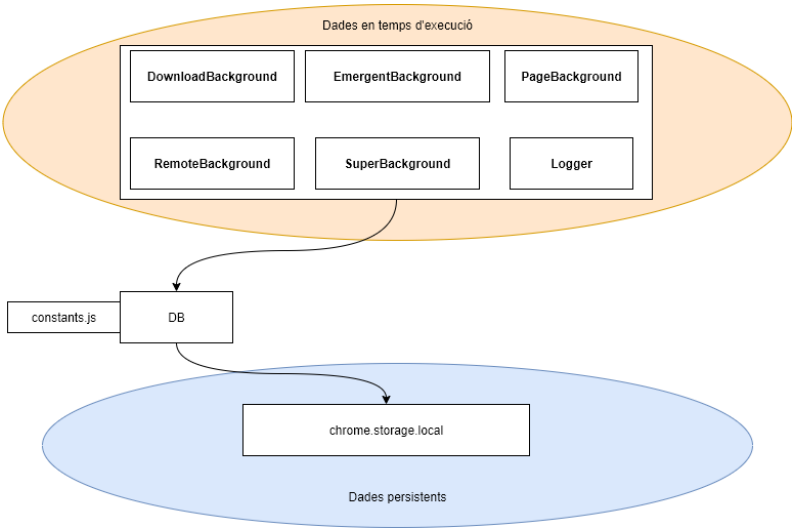


Figura 11: Diagrama de la utilització de dades locals

La figura 11 fa referència a com es gestionen les dades de forma local, per fer-ho totes les classes que ho requereixin utilitzaran el fitxer constants.js, que en aquest cas manindrà un mateix patró d'identificadors per a fer escalable l'extensió i permetre accedir fàcilment a les dades gestionades per un sistema de clau valor proporcionat

per l'API de Chrome, `chrome.storage.local`. Aquestes classes hauran de gestionar internament com volen utilitzar aquestes dades i quan volen guardar-les altre cop al navegador.

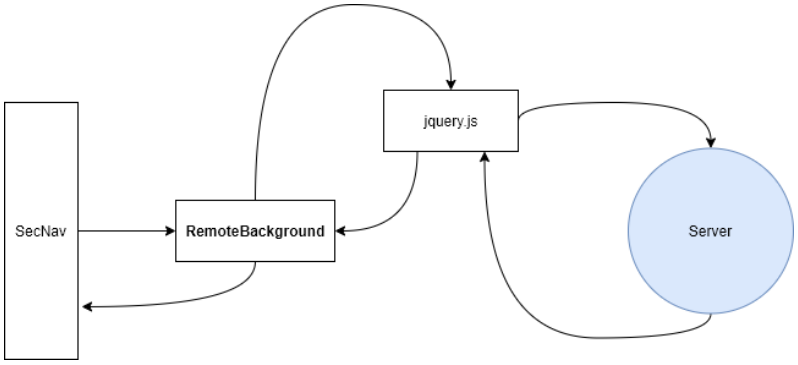


Figura 12: Diagrama de la utilització de dades remotes

La figura 12 fa referència a com es gestionen les dades de forma remota, on es pot veure que tota classe del background que vulgui sol·licitar o enviar dades ho farà a través de la classe `Remotebackground` que al mateix temps utilitzarà el fitxer de Java-Script jquery.js per mitjançant `ajax`, comunicar-se amb el servidor amb el clàssic mètode de sol·licitud i resposta, que s'explicarà en més detall en el capítol d'implementació. El fet que `Remotebackground` intervingui en totes les connexions fa que sigui capaç de detectar errors de connexió i d'identificador i poder actuar en conseqüència per no saturar massa el sistema.

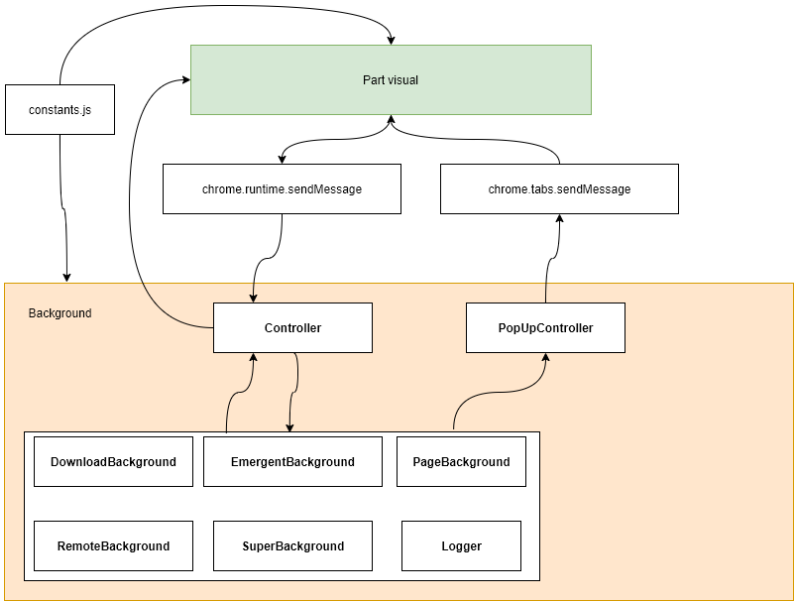


Figura 13: Diagrama de la comunicació entre diferents parts de l'extensió

La figura 13 mostra com es produeix la comunicació entre el background i la part visual. Quan la part visual necessita enviar o rebre dades és el controlador que rep

la sol·licitud i la distribueix a la classe a la qual vagi dirigida. Gràcies a un dels paràmetres de la API de Chrome la part visual rep una resposta amb les dades que s'han considerat necessàries depenent de la sol·licitud.

En cas que sigui el background el que vulgui enviar una sol·licitud, molt probablement informativa, utilitzarà la classe estàtica `PopUpController` que s'encarregarà de enviar el missatge a la pestanya corresponent juntament amb els fitxers necessaris perquè es pugui realitzar la acció que es requereixi.

Tota aquesta comunicació es produeix amb identificadors únics per a cada acció definits corresponentment al fitxer constants.js.

3.3 Configuració

El disseny de Configuració està format pel disseny visual i lògic de l'apartat d'opcions de l'extensió, apart de dissenyar la lògica i la comunicació amb el background s'ha d'escollir una línia de disseny que es segueixi en tots els apartats visuals.

3.3.1 Aspecte visual

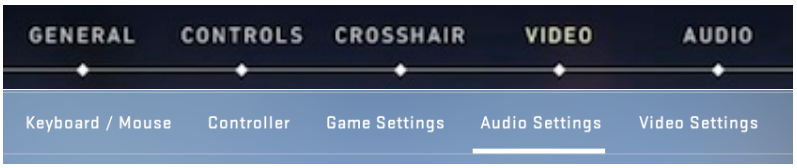


Figura 14: Exemple d'opcions en pestanyes

Pel que fa al disseny de la pàgina d'opcions està basada en panells de configuració que es separen entre si utilitzant pestanyes. Aquesta decisió ve presa al veure la senzillesa d'aquest sistema en diferents programes tal i com es pot observar en la figura 14.

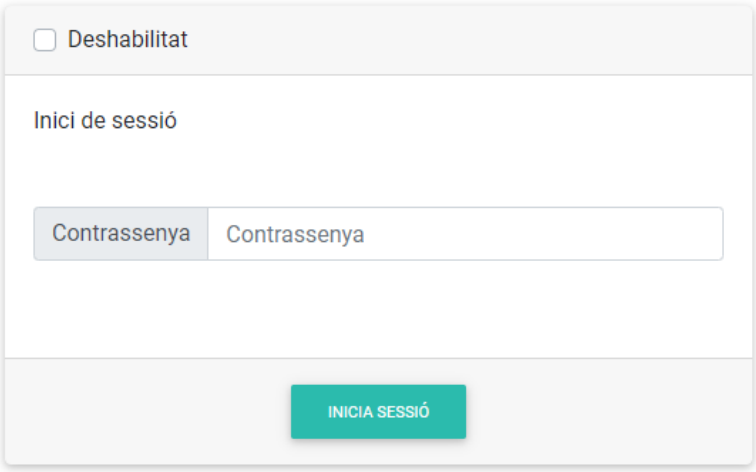


Figura 15: Exemple de targeta

Cada apartat dins de la pestanya es separarà de la resta mitjançant el recurs de la targeta. D'aquesta manera es pot fer cada funcionalitat independent i l'usuari pot veure clarament aquestà separació. (Figura 15)

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	$132,000

Figura 16: Exemple de DataTable

Per mostrar i relacionar-se amb les dades s'ha decidit utilitzar el recurs DataTable (Figura 16) que ve amb el paquet gratuït de MDB i permet un munt d'opcions a una taula HTML que nativament no disposa, aquesta taula serà utilitzada dins de les targetes mencionades anteriorment.

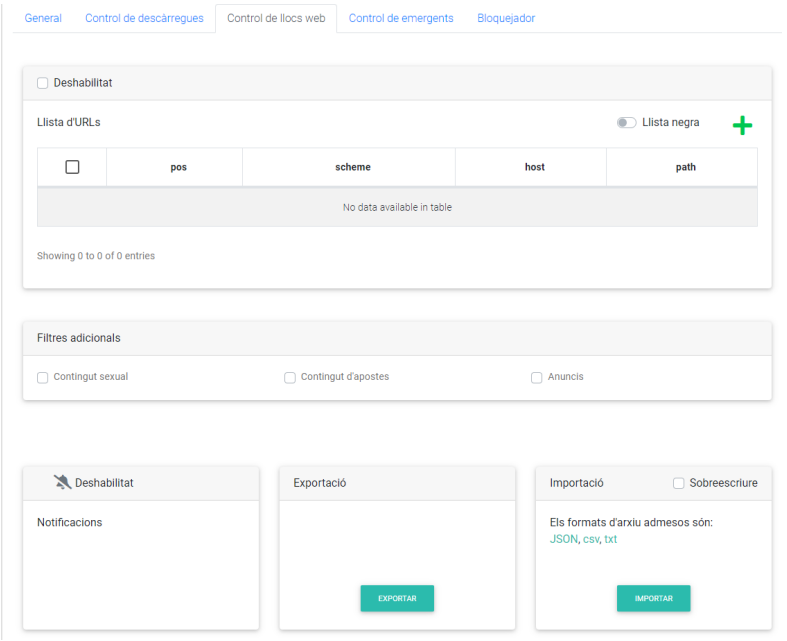


Figura 17: Resultat final

En aquesta Figura 17 es pot veure el resultat final després d'ajuntar els diferents elements escollits pel disseny de la pàgina d'opcions, és una de les pestanyes de l'extensió, es pot veure clarament quina és la pestanya seleccionada i a dins de la mateixa també es pot distingir clarament els diferents apartats que es el que es volia aconseguir amb els recursos escollits anteriorment.

3.3.2 Lògica

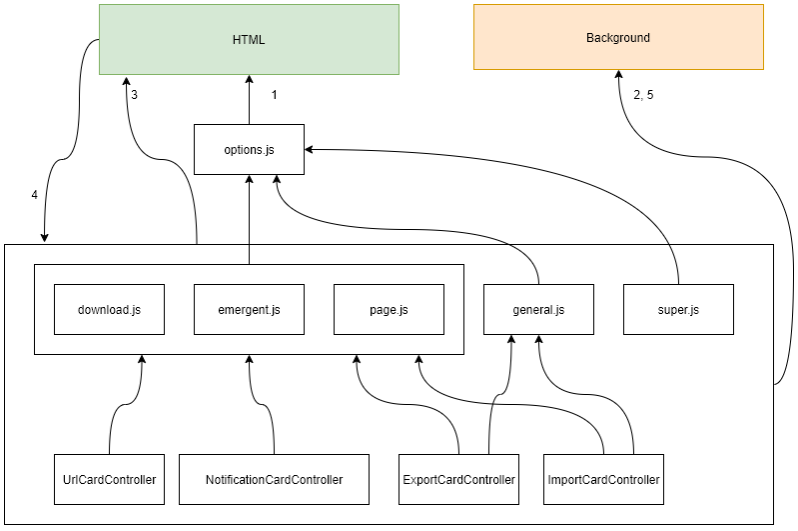


Figura 18: Diagrama de l'apartat de configuració 1

Com es pot comprovar en aquesta Figura 18 al contrari que al background on, majoritàriament, només hi havia classes, en aquest cas tenim més fitxers Java-Script. No val la pena en termes de rendiment fer tantes classes, ja que aquests fitxers s'encarreguen de gestionar cada un la seva secció i tots els esdeveniments derivats de la mateixa, la gestió d'esdeveniments és més eficient en un Java-Script que en una classe pel que fa a les crides.

Les classes que s'han decidit introduir són per controlar elements que són comuns en diferents seccions. Com s'ha pogut comprovar al background tant les seccions de descàrregues, com d'emergents com de pàgines contenien elements en comú, en aquest apartat es segueixen tenint i per tant les classes que gestionen aquests apartats també són comuns. La de gestió d'URL o la de gestió de notificacions són exemples d'apartats comuns.

Pel que fa a les classes d'Exportació i Importació a part de donar servei a les tres seccions anomenades anteriorment, també donarà servei a general.js ja que es permet importar i exportar configuracions senceres de l'extensió.

Cada fitxer s'encarregarà de carregar els elements necessaris al HTML amb les dades necessàries que ells mateixos sol·licitaran al background i de la gestió de les mateixes. Tots els fitxers i les classes tindran accés a background per sol·licitar dades i sol·licitar la modificació de les mateixes, això si, el background sempre té la última paraula sobre si es podrà realitzar o no aquesta modificació.

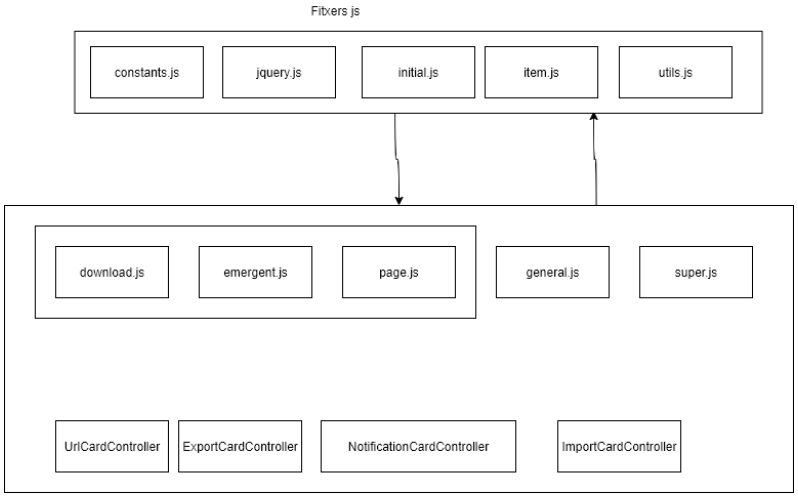


Figura 19: Diagrama de l'apartat de configuració 2

Per carregar l'HTML de la Figura 18 tant els fitxer Java-Script com les classes tindran accés a varis fitxers auxiliars que permeten modificar l'HTML amb components ja configurats o crear-ne de nous d'una manera ràpida i senzilla. D'aquesta manera s'obté un disseny escalable mantenint uns mateixos criteris de disseny en cada apartat encara que es poden modificar per solucionar possibles errors visuals i mantenir una coherència de dimensions.

Per fer sol·licitacions al background disposen de varies utilitats que permeten des de donar-li un format a les dades fins a obtenir els identificadors. (Figura 19)

3.4 PopUp

El PopUp és l'aparta visual que es desplegarà al clicar sobre el logo de l'extensió, una versió molt més simplificada de l'apartat d'opcions.

3.4.1 Aspecte visual

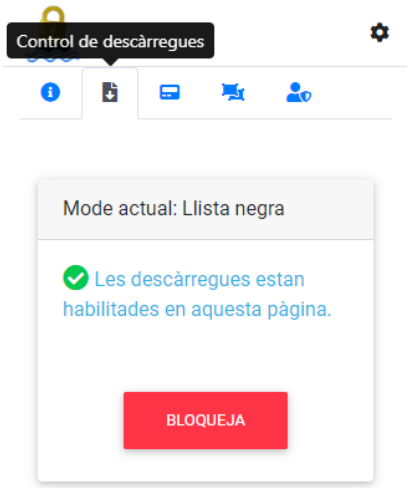


Figura 20: Exemple de pestanyes amb icones

Per a aquest PopUp s'ha realitzat una distribució de pestanyes, en aquest cas en comptes d'estar distingides per text ho fan amb icones en que es pot identificar cada apartat, el text no desapareix, si no que es mostra en forma de 'títol', és com s'anomena aquest recurs al HTML que et permet mostrar text al estar amb el cursor a sobre de l'element. El text està formatat per una altra de les opcions que ens ofereix MDB com és la llibreria popper.js, que ens permet fer que aquests 'títols' siguin més agradables a la vista i es puguin llegir millor.

En el cas de les cartes s'ha optat per minimitzar el màxim el nombre de cartes per secció a ser possible una, d'aquesta manera seguim amb la línia de disseny marcada i s'ajusta a la mida adequada.(Figura 20)

La paleta de colors no s'ha vist afectada.

3.4.2 Lògica

La lògica com era d'esperar, s'ha vist reduïda respecta la pàgina d'opcions. El funcionament però, és exactament igual, la única diferència és que prescindim de les classes que teníem (Figura 18), ja que hi ha variacions en cada secció i en aquest cas s'ha considerat més bona opció no ajuntar conceptes i tractar cada secció per separat.

4 Implementació

La implementació s'ha realitzat utilitzant el llenguatge de programació Java-Script, el llenguatge css i HTML pel que fa a l'extensió. En el cas del servidor s'ha realitzat en python[10] utilitzant l'IDE Pyhcharm[8] de la companyia JetBrains com a editor.

En ambdós casos s'ha utilitzat git[6] per a control de versions.

La implementació es pot dividir en tres apartats, background, visual i servidor. En cada un d'aquest apartats s'utilitzaran els diagrames de flux per explicar-ne la implementació i a partir dels mateixos explicar el per què d'algunes decisions preses.

4.1 Background

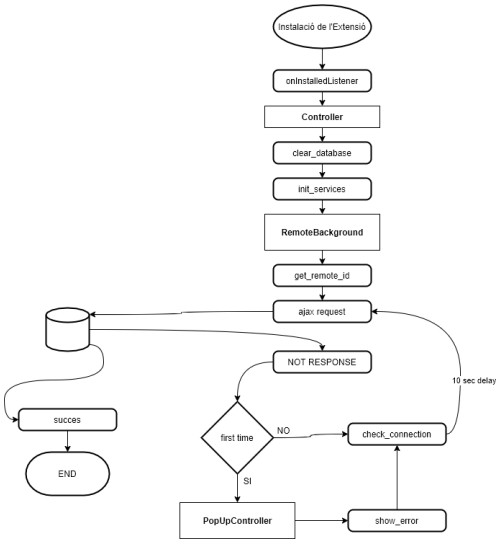


Figura 21: Diagarama d'instal·lació

Quan s'instal·la per primer cop l'extensió es produeix l'esdeveniment gestionat per l'API `runtime onInstalledListener` que cridarà a la instancia de Controller per primer fer una neteja de la memòria del navegador per si quedaven restes d'una instal·lació anterior i iniciarà tots els serveis del background, fent referència a tots els singletons mencionats en el disseny.

Aquest diagrama es centrarà en com s'inicia el `Remotebackground`, ja que té un comportament diferent a la resta. La primera vegada que càrrega les dades, òbviament no hi haurà cap identificador proporcionat pel servidor, per tant és l'extensió qui sol·licitarà un identificador al servidor utilitzant `ajax`. La funció `ajax` és asíncrona per defecte, per tant no bloqueja el flux del programa, si el servidor està disponible es rebrà l'identificador, es guardarà a la memòria del navegador i es donarà per tancada l'acció. En canvi si no hi ha resposta del servidor es notificarà a l'usuari mitjançant `PopUpController` amb una notificació emergent, que en aquest cas igual que en altres, es mostrarà en una pestanya, una nova pàgina HTML en blanc, ja que per qüestions de seguretat Google no permet a les extensions injectar

codi a les pàgines internes de Google, i al ser un missatge important, ja que afecta directament en el comportament de l'extensió, s'ha pres la decisió de donar-li més protagonisme. A part es mostrarà sobre la icona de l'extensió un signe d'exclamació i al passar el ratolí per sobre ens informarà de que no li és possible connectar-se amb el servidor. Un cop notificat l'usuari es repetirà el procés cada 10 segons utilitzant altre cop `ajax` per no aturar l'execució de l'extensió fins que el servidor respongui o es tanqui el navegador. (Figura 21)

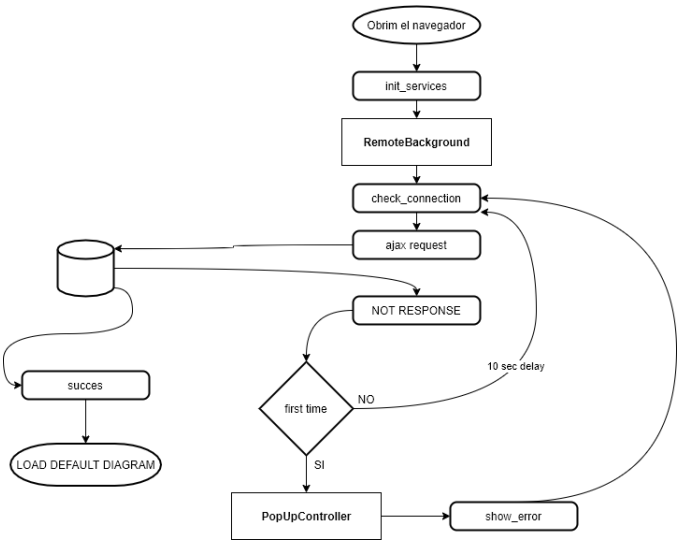


Figura 22: Diagrama de connexió amb el servidor

Quan s'inicia el navegador també s'inicia l'extensió, evidentment d'estar instal·lada i habilitada. Altra vegada el diagrama es centrarà en l'inici de `Remotebackground` que en aquest cas, en comptes de sol·licitar de nou un identificador al servidor simplement farà una sol·licitud per comprovar si el servidor és accessible, s'utilitzarà `ajax` per no aturar el flux d'execució. En cas de que si existeixi una connexió passarà a executar-se el flux explicat en la Figura 5, en cas contrari passaria el mateix que en l'exemple anterior, ja que la funció que s'encarrega de fer aquestes comprovacions és la mateixa (`check_connection`). (Figura 22)

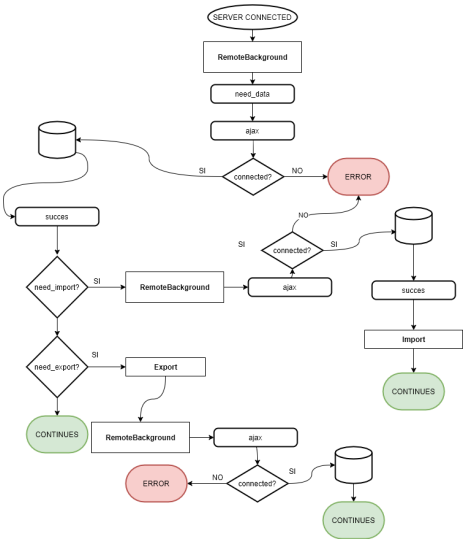


Figura 23: Diagrama de càrrega o descàrrega de dades al servidor

Un cop se sap que l'extensió té connexió al servidor la funció `need_data` mitjançant `ajax` preguntarà al servidor si ha de realitzar alguna acció, el servidor enviarà la resposta de tornada a l'extensió que comprovarà si ha de realitzar una importació de dades des del servidor, una exportació cap al servidor, les dues accions o cap acció. En cas d'haver de realitzar una importació s'enviarà mitjançant `ajax` la sol·licitud per obtenir les dades, un cop es rebin mitjançant la classe estàtica `Import` s'introduiran al sistema. En cas de que sí es requereixi una exportació la classe estàtica `Export` realitzarà l'exportació de dades mitjançant `ajax`, cal destacar que aquesta exportació no contemplarà les dades de la classe `Superbackground` per qüestions de privacitat. Si en qualsevol de les crides `ajax` hi ha un error de connexió, es notificarà l'error i es tornarà al bucle del diagrama anterior del `check_connection`. (Figura 23)

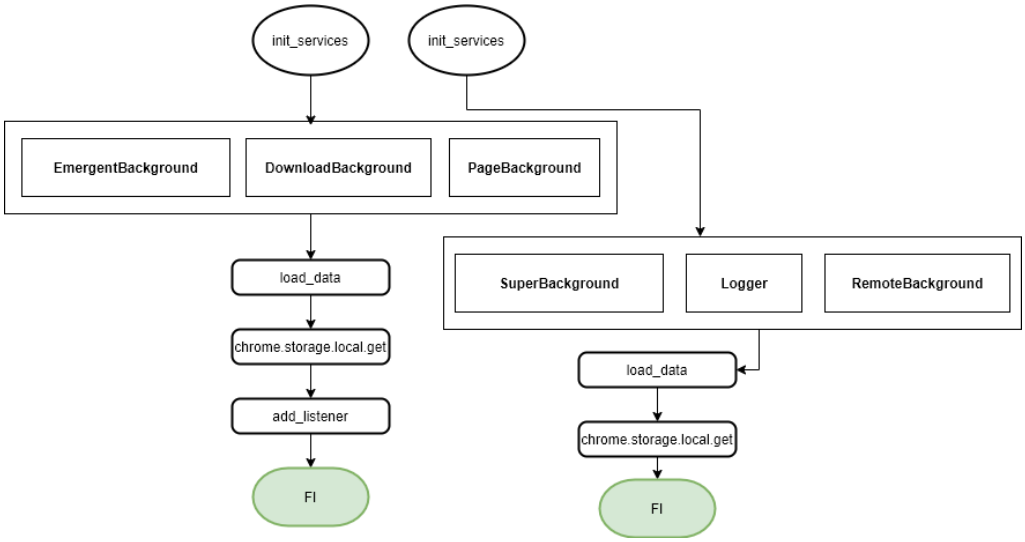


Figura 24: Diagrama d'inicialització de serveis

Aquest diagrama es centrarà en la inicialització dels altres serveis que conté l'extensió, es poden separar en dos grups, el primer està format per, les tres classes que s'han d'encarregar de bloquejar o permetre determinades accions: `Downloadbackground`, `Emergentbackground` i `Pagebackground`. El segon grup està composat per les tres classes que s'encarreguen d'ajudar a les altres tres amb diferents utilitats: `Superbackground`, `Logger` i `Remotebackground`, en aquest últim cas ja s'ha vist en diagrames anteriors que es realitzen més accions, així que aquest estarà centrat en les accions comuns.

El primer grup i el segon comparteixen el 90% de l'execució, primer de tot `init_services` farà una crida al `load_data` de cada una de les instàncies que realitzaran una crida a la API de Chrome *storage*, totes les crides a qualsevol API de Chrome són asíncrones, per obtenir les dades que estan emmagatzemades al navegador,tractar-les en cas que sigui necessari i organitzar-se-les com cada classe trobi adequat. En aquest punt de l'execució el segon grup ja ha acabat, en canvi, el primer ha de realitzar una acció més i és afegir el *listener* de l'API de Chrome corresponent a cada apartat per gestionar els esdeveniments dels que s'ha d'encarregar cada classe. Ara sí tots els serveis estan actius i a punt per realitzar les tasques futures. (Figura 24)

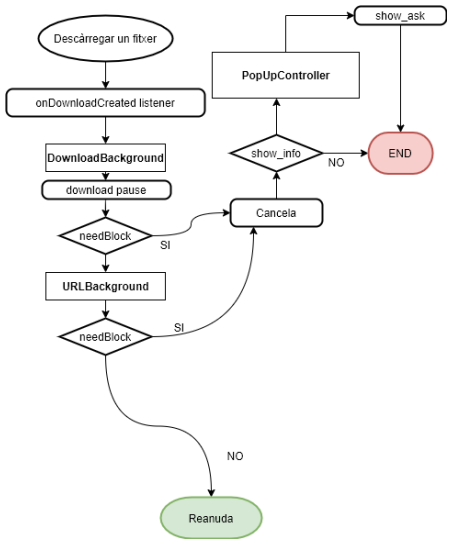


Figura 25: Diagrama de control de descàrregues

Una de les tasques futures que hauran de realitzar els serveis i en aquest cas concret `Downloadbackground`, és la d'analitzar si un lloc web pot o no descarregar un fitxer.

Gràcies al *listener* afegit al iniciar el servei, cada cop que s'inicii una descàrrega aquest ho capturarà i cridarà a `Downloadbackground` perquè la gestioni. El primer que farà `Downloadbackground` serà pausar la descàrrega per analitzar-la sense posar en risc l'ordinador, es comprovarà amb les dades que gestiona localment la extensió si és necessari el bloqueig, en cas afirmatiu es cancel·larà la descàrrega i si l'usuari així ho desitja se l'informarà del bloqueig. Si de manera local no s'ha bloquejat utilitzarà `URLbackground` per decidir si s'ha de bloquejar o no a partir de les dades

remotes. En cas de bloqueig es repetirà el flux de bloqueig i en cas de no requerir bloqueig es reprendrà la descàrrega.

Cal destacar que al contrari del que pot semblar una descàrrega no té una única direcció URL, és a dir, sí, només es descarrega des d'una sola URL, però també tenim el camp *referrer* que fa referència a la URL des de la qual l'usuari ha iniciat la descàrrega, el camp *url* que fa referència la url a on es sol·licita la descàrrega sense redireccions, i *finalUrl* que fa referència a la url des d'on s'està descarregant realment el fitxer. Per tant s'han de comprovar les tres URLs tant en les dades locals si s'escau (pot estar deshabilitat) i a les dades remotes.

En aquest apartat cal explicar el per què de pausar, reprendre i cancel·lar en comptes d'utilitzar un sistema per abans d'iniciar la descàrrega bloquejar-la. La única manera 100% de saber si es vol descarregar algun fitxer és utilitzant un *listener* a `onDownloadCreate`, és veritat que si s'utilitzés l'esdeveniment `webRequestonBeforeSendHeaders` (Figura 26) amb aquests *headers* es podria arribar a esbrinar que es vol descarregar un fitxer, però no es tindrien totes les dades que proporciona l'altre funció que també són utilitzades pel filtre i només es poden aconseguir al iniciar la descàrrega.(Figura 25)

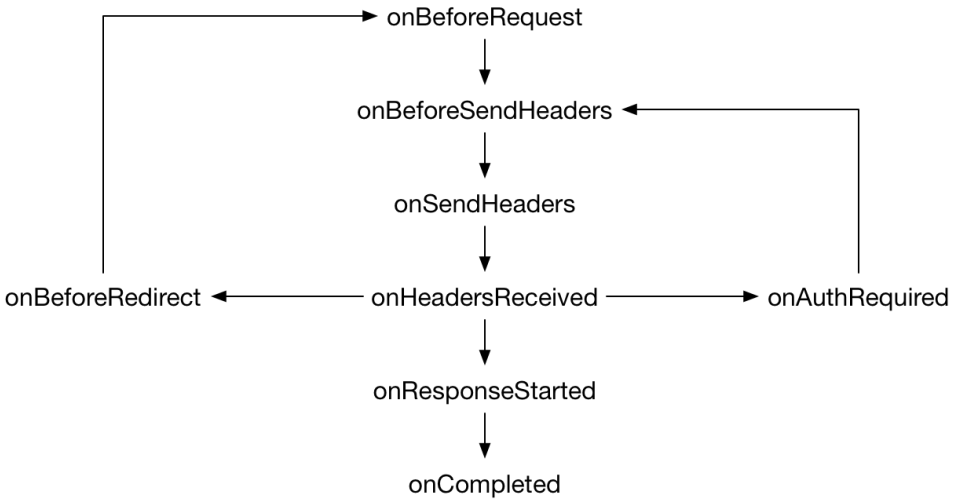


Figura 26: Diagrama dels diferents esdeveniments de *webRequest*

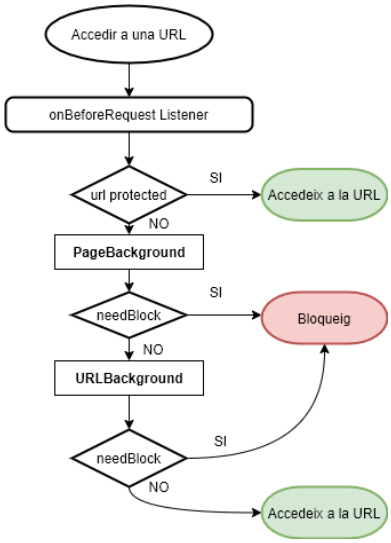


Figura 27: Diagrama de control de pàgines

Abans d'explicar el funcionament alhora de determinar si s'ha de permetre l'accés o no a una pàgina és important remarcar que el *listener* està escoltant a l'esdeveniment `webRequest.OnBeforeRequest`, que tal i com es pot veure en la Figura 26 es realitza abans de que el navegador enviï ni rebi cap dada d'aquesta URL, ja que el que es busca és bloquejar **tota** comunicació amb aquell domini, pàgina, o protocol en cas de bloqueig.

El funcionament de la gestió de pàgines és molt similar al de descàrregues i al d'emergents. Primer de tot el *listener* captura l'esdeveniment, en aquest cas però, abans de passar-li la tasca del bloqueig directament a la classe `Pagebackground` fa una primera comprovació de no bloqueig, ja que hi ha direccions protegides que per l'ús de l'extensió i inclús del navegador no es poden bloquejar, per exemple si es bloqueges la pròpia extensió perdria tot el sentit d'existir igual que si la pròpia extensió bloqueja la connexió al servidor que s'està connectant. Un cop superat aquest primer filtre és el moment de la intervenció de la classe `Pagebackground`.

El funcionament és el mateix que en el control de descàrregues amb la diferència que ni es pausa ni es reprèn ni es cancel·la cap tipus de sol·licitud ja que el control es realitza abans de fer-ne cap. Simplement es determinarà si s'ha d'enviar (no bloqueig) o no s'ha d'enviar (bloqueig) la sol·licitud. (Figura 27)

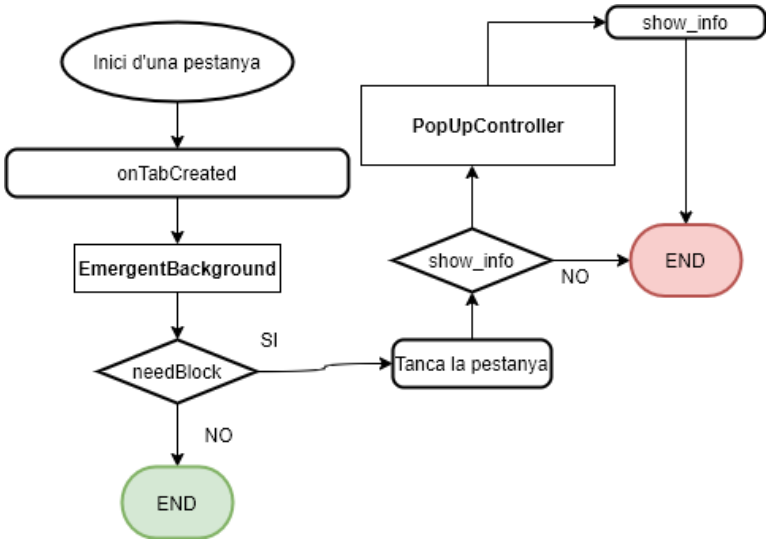


Figura 28: Diagrama de control d'emergents

El comportament de la gestió d'emergents és similar als dos anteriors, encara que en aquest cas no s'utilitzaran dades remotes, ja que s'ha considerat que la gestió d'emergents és una funcionalitat més personal de cada usuari i no pas de perill general. S'ha de recordar que l'extensió volia també servir per a una navegació més còmode i aquest apartat es centra en això. També cal destacar que en cas de voler-ho implementar en futures ocasions requeriria de molt poc esforç en termes de disseny i implementació, ja que la base dissenyada així ho permet. (Figura 28)

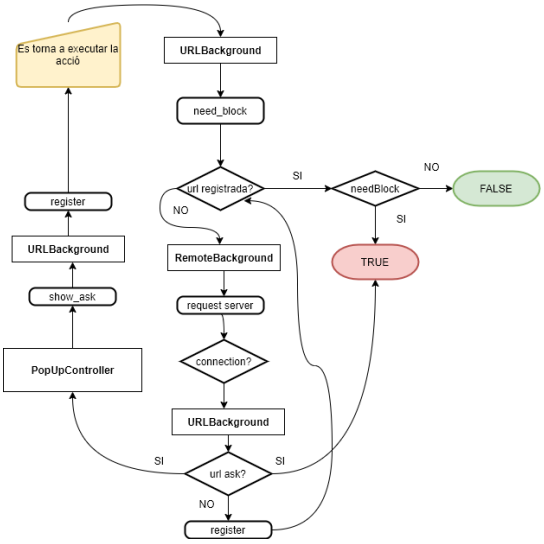


Figura 29: Diagrama de gestió de bloqueig remota

El flux que segueix la funció `need_block` de `URLbackground` s'utilitza en els esdeveniments de control de descàrregues i de pàgines, cal destacar que es podria implementar també en nous serveis que s'incloguin ja que el sistema de bloqueig remot és el mateix en tots els casos, el que canvia en cada apartat és l'acció final

després de la decisió presa, la decisió sempre serà o bloquejar o no bloquejar.

Quan es fa la crida al `need_block` primer de tot es comprova si aquesta URL ja ha estat registrada, en cas afirmatiu es comprova l'estat d'aquesta URL i es retorna el resultat. En cas negatiu es fa ús de la classe `Remotebackground` per sol·licitar les dades al servidor.

És important conèixer que aquesta sol·licitud no és asíncrona, si no que és síncrona ja que el que es vol és aturar el flux d'execució i només bloquejar quan es sàpiga segur que s'ha de bloquejar, és a dir no es pot deixar a l'esdeveniment de tornada d'`ajax` perquè s'ha de prendre una decisió sobre si bloquejar o no en aquell precís moment. El contra és que pot alentir el funcionament de l'equip, és per això que `Remotebackground` comprova a cada sol·licitud si hi ha connexió per en cas de que s'hagi perdut suspendre totes les crides de sol·licituds sortints fins que es restableixi. En cas que `Remotebackground` detecti que no hi ha connexió el `need_block` de `URLbackground` no bloquejarà cap URL que no tingui registrada, ja que no podrà disposar de les dades sense una connexió amb el servidor.

En cas que no hi hagi cap problema i es puguin rebre les dades del servidor es comprovarà si se l'hi ha de preguntar a l'usuari sobre el futur de l'acció , en cas que se l'hi hagi de preguntar es retornarà que s'ha de bloquejar. Quan respongui es guardarà la resposta i es tornarà a realitzar la acció que havia estat bloquejada en un primer moment. Si l'usuari no ha de respondre es registrarà la informació aportada i es tornarà al punt de si la URL està registrada. (Figura 29)

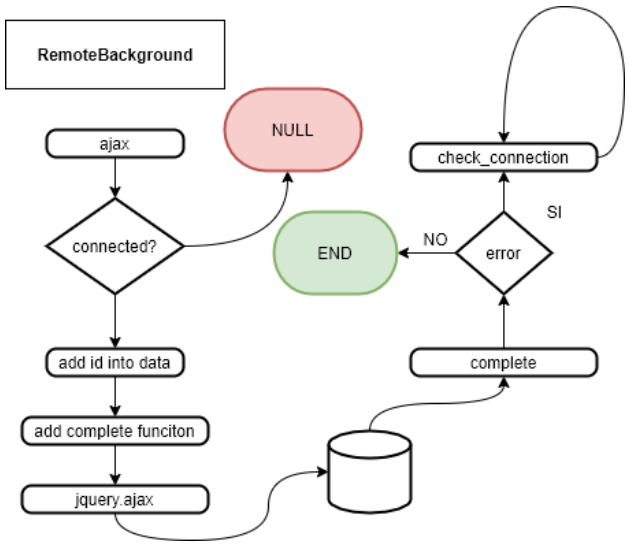


Figura 30: Diagrama d'ajax de Remotebackground

La funció `ajax` de `Remotebackground` comprova abans de fer cap altre acció si hi ha una connexió establerta amb el servidor, en cas negatiu retorna `null` i en cas afirmatiu afegeix l'identificador obtingut en la figura 21 a les dades. També afegeix la funció `complete` que s'executarà al finalitzar-se la petició. Aleshores s'executa la funció `ajax` de jquery. Un cop completada la sol·licitud s'executarà la funció `complete` i comprovarà si la connexió segueix vigent o ha de reconnectar-se.

(Figura 30)

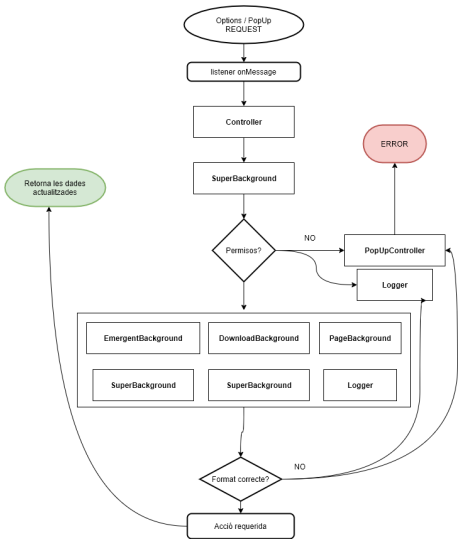


Figura 31: Diagrama de control de sol·licituds

Pel que fa a la comunicació entre la part visual i el background centrada amb el background, s'utilitzarà la API de Chrome `runtime` afegint un *listener* al esdeveniment `onMessage` que l'hi passarà la sol·licitud al controlador que al mateix temps l'hi passarà a la classe `Superbackground` perquè comprovi si es disposen dels permisos necessaris per a dur a terme l'operació. En cas de no disposar d'aquests permisos es notificarà a l'usuari mitjançant `PopUpController`. En cas de disposar dels permisos necessaris es farà la crida al servei que el controlador havia decidit que anava destinada aquella sol·licitud, aquest servei comprovarà si la sol·licitud és correcte o no. En cas afirmatiu es realitzarà la acció requerida i posteriorment es retornaran les dades a vista. En cas contrari es notificarà a l'usuari sobre l'error que hi ha hagut. (Figura 31)

S'ha de destacar que la classe `Logger` sempre que es produeix un error o es realitza una acció determinant com podria ser el bloqueig d'una pàgina rep un *log*, no s'ha inclòs en els diagrames anteriors ja que és una cosa constant i repetitiva i no és amb el que es centra l'objectiu dels diferents diagrames, si no que busquen que s'entenguin les operacions principals que fa l'extensió.

Pel que fa al seu comportament, al rebre un *log* es mira si es vol escriure en el *log* que pot consultar l'usuari o en el de desenvolupador. S'escriu al principi de la resta de *logs* guardant l'hora actual i el missatge, que acostuma a estar inclòs en els *locales* per tal de poder llegir la informació en el llenguatge del navegador. També pot incloure contingut no traduïble com dades d'un error d'execució o URLs entre d'altres. Un cop s'ha escrit es guarda a la memòria del navegador i en cas de tractar-se del *log* de desenvolupament s'envia al servidor utilitzant `Remotebackground` i `ajax`. Com a cada ús d'`ajax`, si hi ha un error de connexió es tornarà a intentar restablir la mateixa. (Figura 32)

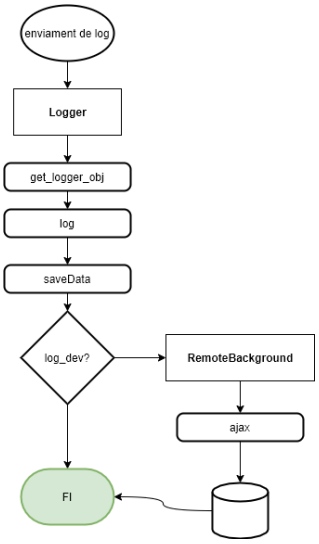


Figura 32: Diagrama de gestió de *logs*

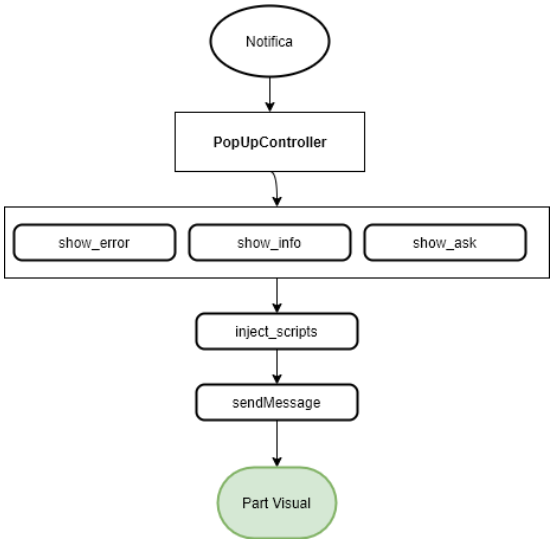


Figura 33: Diagrama de enviament de PopUps

Per enviar les diferents notificacions a l'usuari la classe `PopUpController` quan rep el missatge que ha d'enviar a la part visual de l'extensió la funció cridada injecta els .js i els .css que necessitarà per mostrar el missatge i utilitzant la funció del'API de Chrome `tabs` s'envien les dades a la vista. (Figura 33)

4.2 Visual

4.2.1 *Locales*

Pel que fa als idiomes disponibles per l'extensió són el català, castellà i anglès. L'extensió depenent de en quin idioma estigui la interfaç de Chrome utilitzarà un

idioma o un altre, en cas que Chrome estigui en un idioma diferent als mencionats per defecte es mostrarà en anglès.

Per canviar l'idioma de Chrome s'ha d'accedir a: *opcions* →opcions avançades →idiomes →idioma →més opcions →mostra Chrome en aquest idioma.

4.2.2 Aspecte Visual

En aquest apartat s'explicarà i mostrarà les diferents seccions de configuració amb una breu explicació dels seus elements.

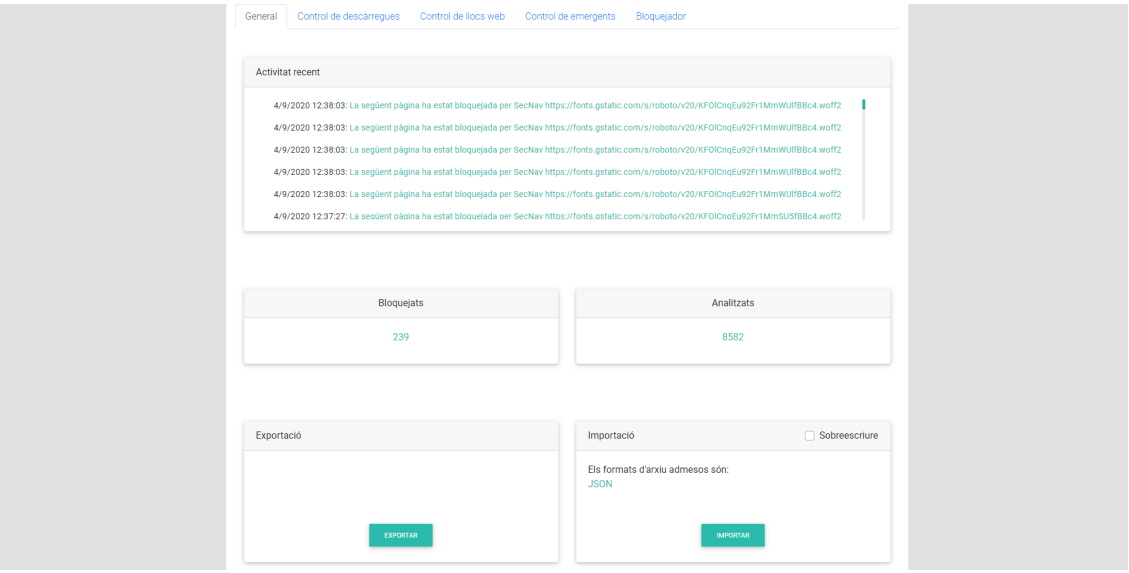


Figura 34: Secció general de configuració

La pestanya general de la pàgina de configuració conté 5 targetes, s'ha optat per una distribució 1-2-2.

La targeta activitat recent necessita d'una gran amplitud perquè els últims registres que mostra es puguin llegir d'una manera més clara. Es mostra l'hora d'un color negre perquè no destaqui i el missatge principal amb el color color-default de la paleta de colors.

En canvi les targetes d'accions bloquejades i d'accions analitzades no necessiten tanta amplada i estan relacionades entre elles, així que s'ha decidit ajuntar-les a la mateixa línia.

Per últim les targetes d'exportació i importació, que són comunes a gairebé totes les seccions, no necessiten de molta amplada i estan relacionades. La carta d'importació aprofita la capçalera de la targeta per incloure el botó de sobreescriure sense ocupar espai extra. Cal anotar que perquè les dues cartes tinguin les mateixes dimensions s'ha hagut de realitzar una modificació extra per Java-Script, ja que una contenia text i l'altre no. (Figura 34)

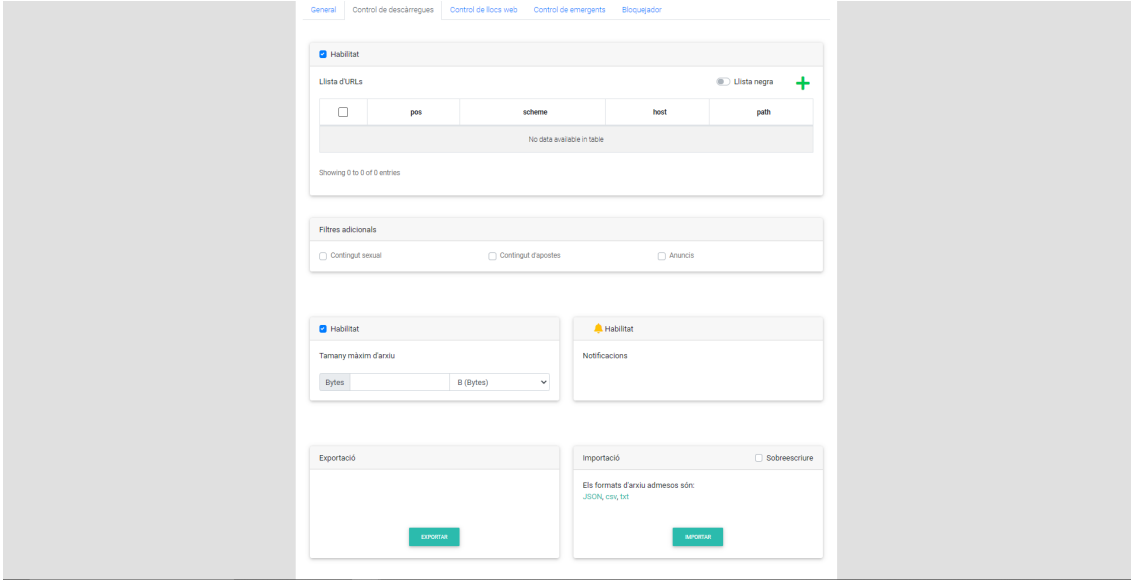


Figura 35: Secció de gestió de descàrregues de configuració

La pestanya de gestió de descàrregues conté 6 targetes, s'ha decidit utilitzar una distribució de 1-1-2-2.

La targeta llista d'URLs, comú tant a la secció de gestió de llocs web com a la de gestió d'emergents, requereix d'una gran amplada ja que utilitza una taula amb 5 columnes i com més ample més llegible serà per l'usuari. Aquesta targeta inclou a la seva capçalera un 'checkbox' per habilitar o deshabilitar la funció, un 'switch' a la part dreta del títol per canviar el mode d'utilització i just al seu costat un botó per afegir nous elements. La primera columna també és un 'checkbox' que, en aquest cas, ens permetrà seleccionar les fileres per poder-les eliminar gràcies a un botó en forma de paperera que apareixerà al costat del botó d'adició. Destaca l'ús del color color-success per el signe d'afegir, ja que el verd normalment es relaciona amb les accions positives, i l'us del color color-danger pel botó d'eliminació, ja que el vermell destaca més i es sol prestar més atenció.

La targeta filtres addicionals, com el seu nom indica, permet seleccionar filtres addicionals pel servidor remot. S'ha optat per mantenir-la sola en una fila, ja que encara que no necessiti tanta amplitud amb la meitat d'una fila no és suficient per mantenir els tres 'checkbox' en una sola filera. Aquesta targeta és comú amb la pestanya gestió de llocs web.

Les targetes tamany màxim d'arxiu i notificacions no tenen res en comú per estar a la mateixa fila, excepte l'amplada necessària. Ambdues inclouen la opció d'habilitar o deshabilitar a la capçalera amb la diferència de que en el cas de la de notificacions conté una campana amb el color color-alert, que al deshabilitar-la es torna gris. Aquesta targeta és comú a les mateixes seccions que la targeta llista URLs. La targeta tamany màxim d'arxiu utilitza el mateix 'checkbox' que la targeta de llista d'URLs. (Figura 35)

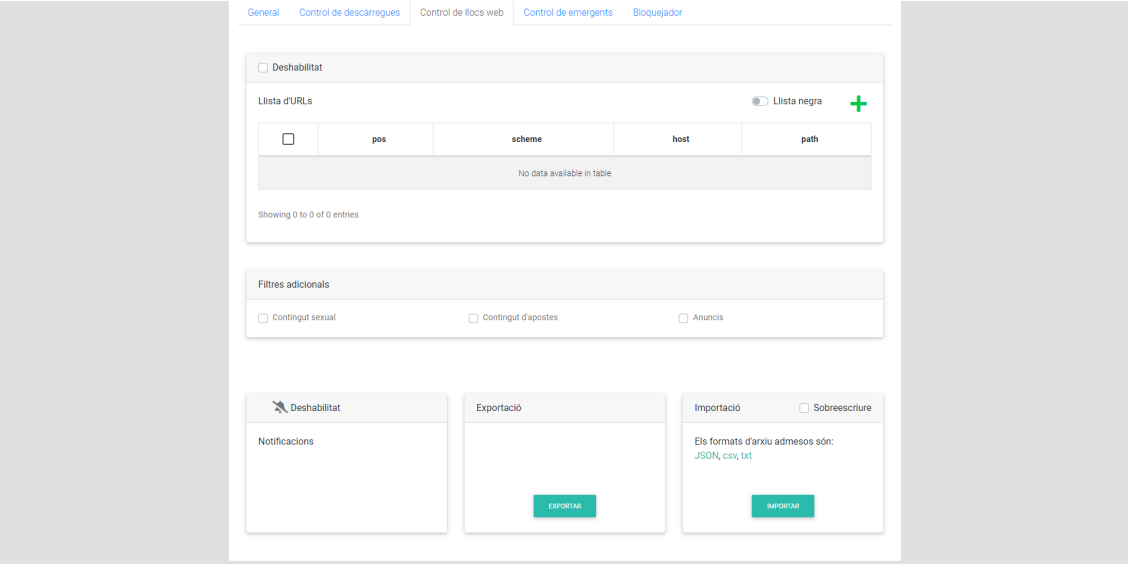


Figura 36: Secció de gestió de llocs web de configuració

La pestanya de gestió de descàrregues conté 5 targetes, que s'organitzen en un 1-1-3.

Totes les targetes apareixen descrites en els apartats anteriors, només comentar que en aquest cas s'opta per ajuntar a la mateixa filera la carta de notificacions, exportació i importació perquè la amplada mínima de les tres cartes ho permetia. (Figura 36)

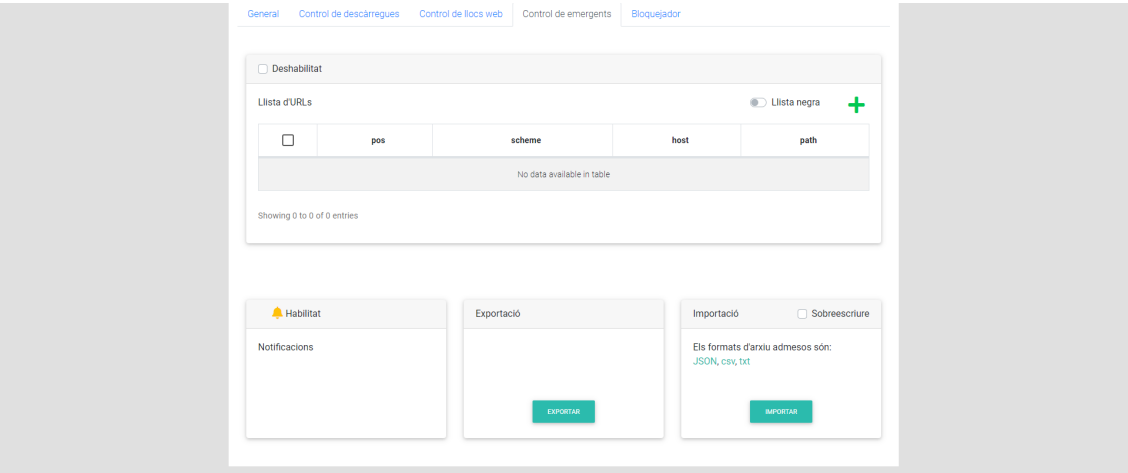


Figura 37: Secció de gestió d'emergents de configuració

La pestanya de gestió d'emergents conté 4 targetes, disposades en un 1-3.

Aquesta pestanya conté les mateixes targetes que la pestanya de gestió de llocs web, excepte per la de filtres addicionals, que com ja s'ha comentat en la implementació del background, s'ha decidit no comptar amb aquesta funcionalitat. Ja que la targeta no inclosa ocupava una fila no s'ha hagut de reajustar la distribució de la resta de targetes. (Figura 37)

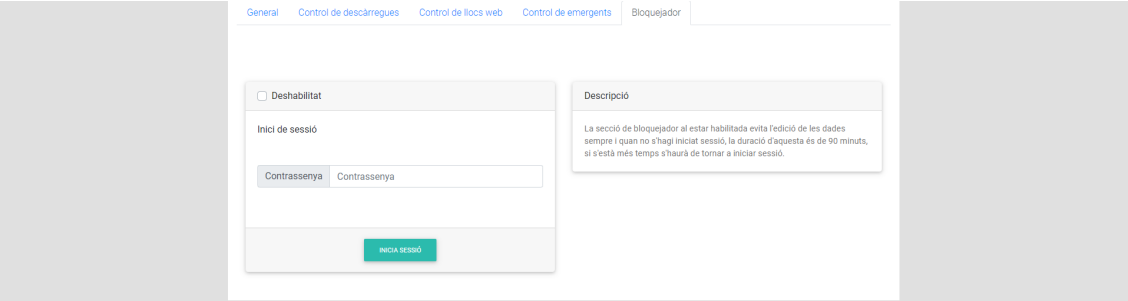


Figura 38: Secció bloquejador de configuració

La pestanya bloquejador és la que menys targetes conté, concretament 2 i estan disposades en una fila de 2.

La targeta de descripció s'ha decidit incloure en aquesta pestanya perquè podia arribar a generar confusions, ja que no ha estat possible fer entendre de manera intuïtiva quina era la seva funcionalitat.

La targeta inici de sessió torna a utilitzar el 'checkbox' utilitzat ja en varies targetes vistes prèviament, és la primera targeta en utilitzar el peu de la targeta. Aquest inclou un botó amb el color color-default com tots els altres botons analitzats.

Cal destacar que depenent de l'estat de l'usuari, és a dir si s'ha registrat, si té la sessió iniciada o no, canvia el contingut de tota la targeta, excepte la capçalera, per a poder realitzar les diferents accions com registrar-se, iniciar sessió (estat actual) o canviar la contrasenya. (Figura 38)

4.2.3 Lògica

Aquest apartat té una implementació que en tots els apartats té la mateixa estructura, encara que, evidentment, es realitzen accions diferents.

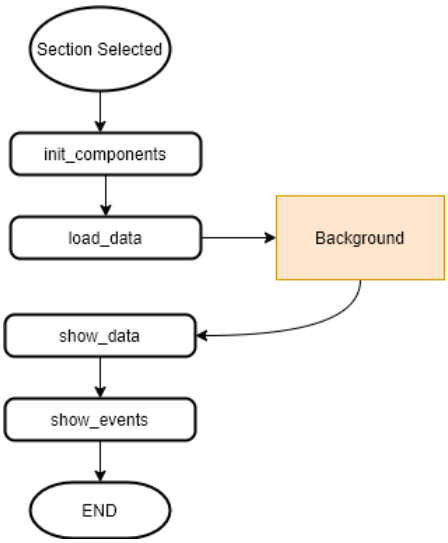


Figura 39: Diagrama de secció visual

Sempre que es selecciona una secció tant de la pàgina de configuració com la del pop-up s'inicien els components HTML generats per Java-Script utilitzant les utilitats creades i les que aporten llibreries addicionals, en aquest cas jquery.

Aquesta acció es realitza únicament la primera vegada que es carrega la pestanya, aleshores es fa una sol·licitud al background mitjançant l'API de Chrome `runtime` mencionada anteriorment amb l'identificador de la sol·licitud que es voldrà realitzar. Un cop les dades han estat enviades per part del background es carreguen els components inicialitzats prèviament amb les dades actualitzades i s'afegeixen *listeners* per a cada esdeveniment que es necessiti controlar. (Figura 39)

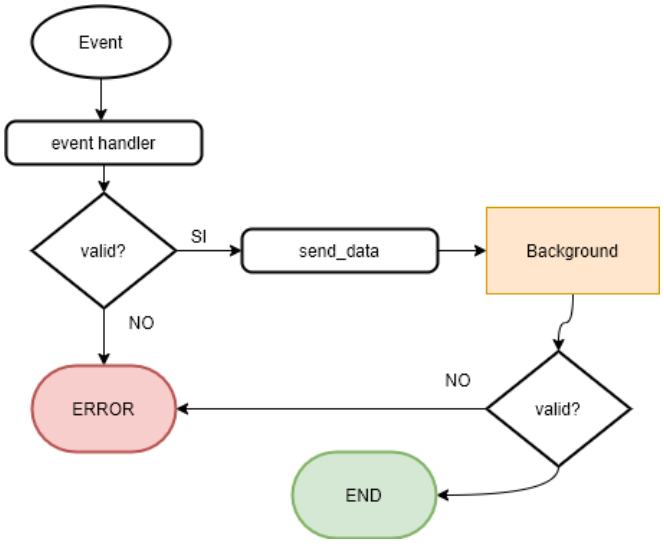


Figura 40: Diagrama de gestió d'esdeveniments visuals

Quan es produeix un esdeveniment, el *listener* afegit comprovarà si les dades són vàlides. En cas afirmatiu les enviarà al background mitjançant `runtime.sendMessage` que les tornarà a comprovar com hem vist a la Figura 31 si tot ha estat correcta acabarà aquesta execució. En cas de qualsevol dels dos errors s'informarà de l'error. Evidentment cada esdeveniment tindrà una comprovació diferent i les sol·licituds seran enviades a la secció del background que correspongui. (Figura 40)

S'ha de tenir en compte que encara que es mantingui la mateixa estructura cada funció s'ha implementat tenint en compte les necessitats de cada secció tant a nivell visual com a nivell lògic.

Com s'ha vist en el disseny en aquest apartat hi ha d'haver 4 classes, s'ha de comentar un canvi respecte la implementació de les classes del background a nivell de codi:

```
class Urlbackground {
    constructor(db) {
        this.urls = undefined;
        this.urls_remote = [];
        this.urls_session = [];
        this.urls_block = [];
```

```
        this.urls_filters = [];
        this.enabled = undefined;
        this.type = undefined;
        this.dB = db;
    }

function UrlCardController(section, dB, filters = true) {
    this.section = section;
    this.charged = false;
    this.dB = dB;
    this.filters = filters;
    that = this;
}
```

Per aquesta comparativa s'han agafat classes "equivalents"però una de background i la altre de la pàgina d'opcions. Com es pot observar hi ha una diferència òbvia i és la manera de declarar les classes. El per què d'aquesta decisió ve degut a que a Java-Script, per temes d'herències, es comporten de manera diferent les dues declaracions a un nivell més intern.

En el cas de declarar-ho com a *class* quan es crea un objecte que hereta d'un altre es podria dir que es creen dos instàncies una pare l'altre filla i fa que alhora d'afegir funcionalitats sigui més estable. En canvi fer-ho amb la forma *function* es podria dir que és més eficient ja que realment només crea una instància, però com més complexa sigui la classe més probabilitats d'error hi ha. Com al background es treballa amb herències s'ha escollit la forma de *class* per evitar possible problemes futurs i a la pàgina d'opcions no estava previst utilitzar herències per tant s'ha decidit utilitzar aquesta altra forma d'implementació que a priori sembla tenir una millor eficiència a nivell de còmput.[5]

4.3 Servidor

El servidor en aquest treball està orientat a nodrir l'extensió i a nodrir-se ell mateix a través d'ella, és un element que està pensat per treure a la llum el potencial d'una extensió d'aquestes característiques.

No es vol dissenyar i implementar un servidor de moltes característiques, si no que el que es busca és un servidor senzill capaç de manejar bases de dades amb facilitat i capaç de funcionar com una API de rebre sol·licituds i enviar dades de forma senzilla. Tenint en compte totes aquestes necessitats s'ha decidit utilitzar Django, un framework de Python que està destinat a aplicacions web, conjuntament amb Django REST framework, un framework de Django per facilitar la infraestructura d'una API.

Django ofereix una gran facilitat de manejar bases de dades, crear endpoints de manera senzilla i eficaç i té un munt de complements que faran la implementació d'una API més senzilla. A més la utilització de Python com a llenguatge de programació fa que sigui senzill realitzar noves funcionalitats a partir de dades obtingudes

ja que disposa de moltes llibreries orientades a ànalisi de dades.

El servidor per defecte correrà al localhost concretament a la IP 127.0.0.1 i port 8000 .

Tenint això en ment la implementació del servidor es distribuirà en tres apartats, la base de dades, els *endpoint* i la lògica.

4.3.1 Base de dades

Ja que el servidor no pretén ser l'element principal i serà creat com a demostració, com a base de dades s'utilitzarà sqlite, la base de dades per defecte de django que no és la més potent ni molt menys, hi ha motors com postgress o sql server o mysql que es desenvoluparien molt millor. Però seria una configuració i una instal·lació extra que no és necessària ja que es treballarà amb dades d'exemple d'un volum baix. D'altra banda és molt més portable i en cas de realitzar exemples en múltiples dispositius i múltiples xarxes és molt portable.

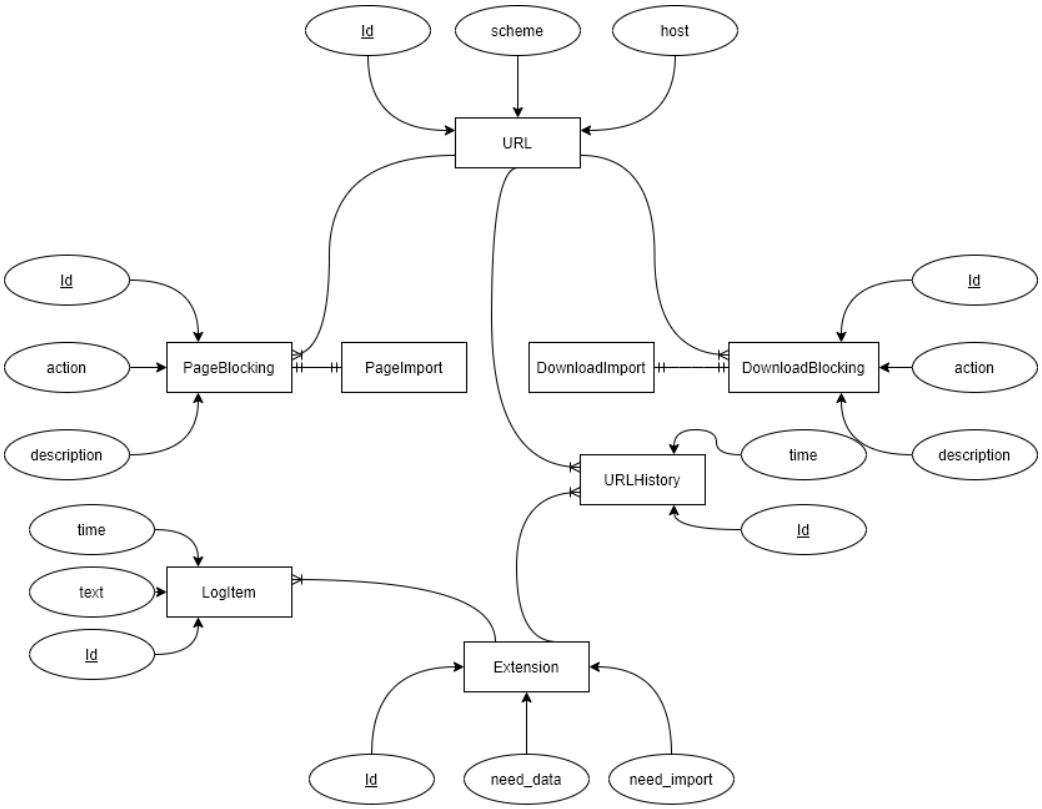


Figura 41: Diagrama d'entitats de la base de dades

En aquest diagrama es pot veure com es disposa d'una entitat URL, composada per un id, protocol i domini la qual servirà per alimentar tant 'PageBlocking' com 'DownloadBlocking' que estan composades per un id, acció i descripció. Encara que tinguin els mateixos atributs la descripció té característiques internes diferents. Cada una respectivament tenen una relació 1 a 1 respecte 'PageImport' i 'Downloa-

dImport' aquestes dues entitats estan orientades a les importacions que pot realitzar la extensió a petició del servidor.

Per l'altra banda també es disposa de l'entitat 'Extension' que conté un id, que en aquest cas serà de 32 caràcters, un camp need_data i un altre need_import que determinaran si l'extensió necessita enviar o sol·licitar dades.

LogItem i URLHistory s'encarreguen de emmagatzemar les dades rebudes de l'extensió LogItem conté un id, text i la data en què s'ha obtingut, per l'altre banda URLHistory té un id, la data en què s'ha obtingut, però també la URL. (Figura 41)

4.3.2 *Endpoints*

El servidor tindrà un total de set *endpoints* dedicats a la connexió amb l'extensió i dos dedicats a la seva instal·lació i actualització:

- **get_id/:** És l'endpoint definit al fitxer constants.js de l'extensió on sol·licitarà el seu id.
- **pages/:** És l'endpoint definit al fitxer constants.js de l'extensió on anirà a buscar les dades per permetre o no l'accés a una pàgina.
- **downloads/:** És l'endpoint definit al fitxer constants.js de l'extensió on anirà a buscar les dades per permetre o no la descàrrega d'un fitxer.
- **load/:** És l'endpoint definit al fitxer constants.js de l'extensió on sol·licitarà si ha de realitzar alguna importació o exportació de les dades.
- **default/:** És l'endpoint definit al fitxer constants.js de l'extensió on sol·licitarà les dades a importar.
- **log/:** És l'endpoint definit al fitxer constants.js de l'extensió on anirà enviant els log de desenvolupador.
- **export/:** És l'endpoint definit al fitxer constants.js de l'extensió on enviarà les dades exportades.
- **ext/update.xml:** És l'endpoint des d'on es podrà actualitzar l'extensió.
- **ext/crxfile.crx:** És l'endpoint des d'on es podrà descàrregar el fitxer crx l'extensió, és a dir el d'instal·lació.

D'altra banda per a facilitar l'accés a la base de dades utilitzarà els endpoint d'admin renombrats a usuarisbase, ja que admin no és un bon nom per tenir recursos d'administrador, des d'on es podrà fer una gestió molt bàsica de la base de dades:

- **usuarisbase/:** És l'endpoint d'administrador de d'on es pot gestionar la base de dades.

4.3.3 Lògica

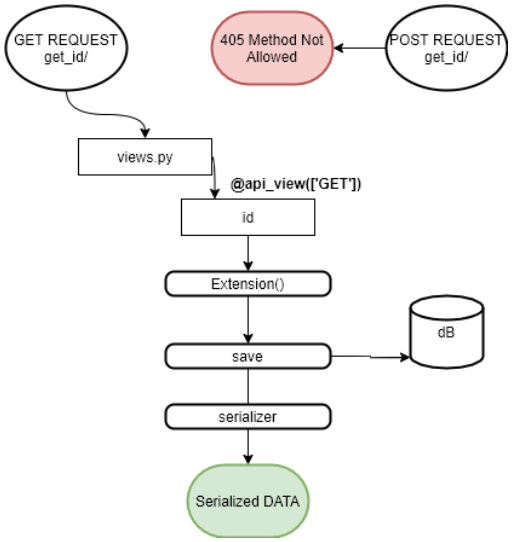


Figura 42: Diagrama de flux de la sol·licitud d'id

Quan el servidor rep una sol·licitud al endpoint get_id/, primer de tot es comprova que la sol·licitud sigui un 'GET', en cas de ser un 'POST' la resposta del servidor serà un error 405 Method Not Allowed, aquesta comprovació es fa internament gràcies al decorador `@api_view(['GET'])` de la llibreria Django REST framework, incorporat a la funció `id` que s'encarrega de gestionar la sol·licitud d'aquest endpoint.

La funció simplement crea un objecte Extension del model de Django, el guarda a la base de dades, serial·litza les dades per a ser enviades en un format JSON i envia aquestes dades serial·litzades. (Figura 42)

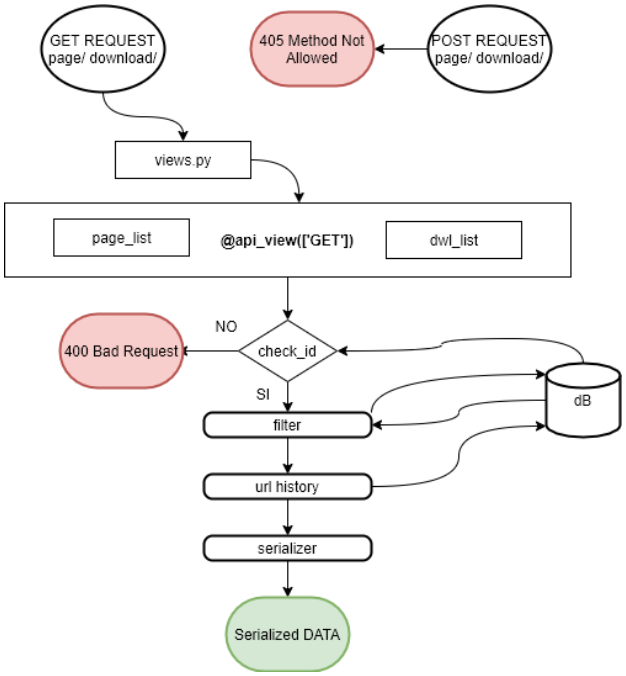


Figura 43: Diagrama de flux de la sol·licitud de gestió de llocs web i de descàrregues

El servidor al rebre la sol·licitud a l'endpoint page/ o a l'endpoint download/ realitza la mateixa comprovació del mètode de la sol·licitud, ja que tant les funcions `page_list` i `dwl_list` que gestionaran la sol·licitud també incorporen el decorador `@api_view(['GET'])`.

Primer de tot es comprova si la sol·licitud inclou l'id de l'extensió i si aquesta correspon a alguna registrada a la base de dades, en cas de que no sigui així el servidor respondrà a la sol·licitud amb un codi 400 Bad Request. Si no hi ha cap problema amb l'id es realitzaran els filtres pertinents respecte el model de dades que es vulgui tractar, llocs web o descàrregues, juntament amb el domini que ha d'aportar la sol·licitud i si s'escau els filtres addicionals. Es guarda a la base de dades la sol·licitud feta per l'extensió abans de serial·litzar les dades filtrades anteriorment i enviar-les. (Figura 43)

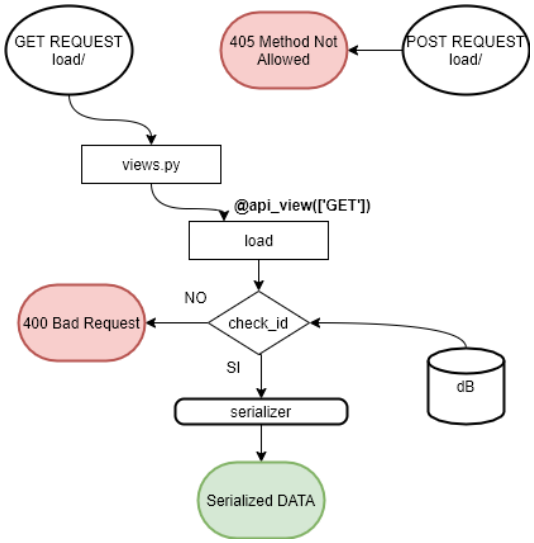


Figura 44: Diagrama de flux de la sol·licitud per si és necessària la importació o l'exportació de dades

En el moment que arriba la sol·licitud a l'endpoint load/ es torna a repetir la comprovació dels diagrames anteriors gràcies a `@api_view(['GET'])`.

En aquest cas el flux és molt més curt, ja que la informació requerida està el mateix model extensió, per tant en cas de no poder carregar l'objecte a través del `check_id` es respondrà amb el codi 400 Bad Request. En cas contrari es serial·litzaran les dades i es respondrà a la sol·licitud amb les dades serial·litzades. (Figura 44)

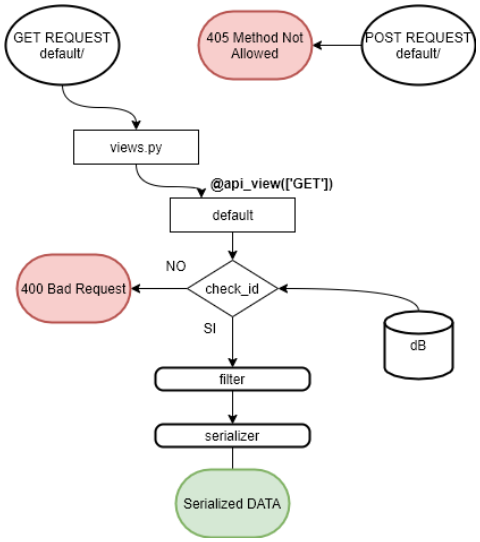


Figura 45: Diagrama de flux de la sol·licitud per la importació de dades

Aquesta sol·licitud manté l'esquema de les anteriors, cal destacar en aquest cas que per defecte els filtres vindran donats pels models de dades PageImport i DownloadImport, encara que es pot canviar per a configuracions predeterminades, inclús

si es sap l'id concret d'una extensió es poden prendre decisions diferents depenent d'aquest identificador, és a dir per defecte es realitza una acció però al canviar-la, sempre i quan es mantingui una coherència de camps, es poden aconseguir resultats molt diferents. (Figura 45)

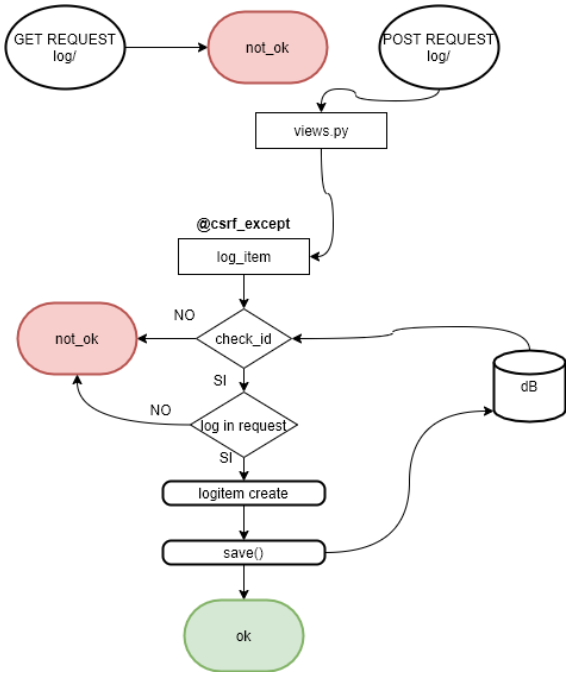


Figura 46: Diagrama de flux de l'enviament dels log

Quan l'extensió envia les dades de log, el servidor, al contrari que en els casos anteriors no fa una comprovació del mètode per defecte, ja que per agilitzar les sol·licituds la funció `log_item` incorpora el decorador `@csrf_except`, que fa que no sigui necessari en un post el camp `csrf` obligatori per defecte i imprescindible per al decorador `@api_view`, per tant es realitza per codi i per això canvia la forma de la resposta.

Just després de la comprovació del mètode es revisa l'identificador i si en les dades aportades s'inclou el log, en cas negatiu es retorna un 'not_ok'. I pel que fa al cas afirmatiu es crea un objecte LogItem del model relacionat amb l'extensió que ha enviat les dades es guarda a la base de dades i es retorna un 'ok'. (Figura 46)

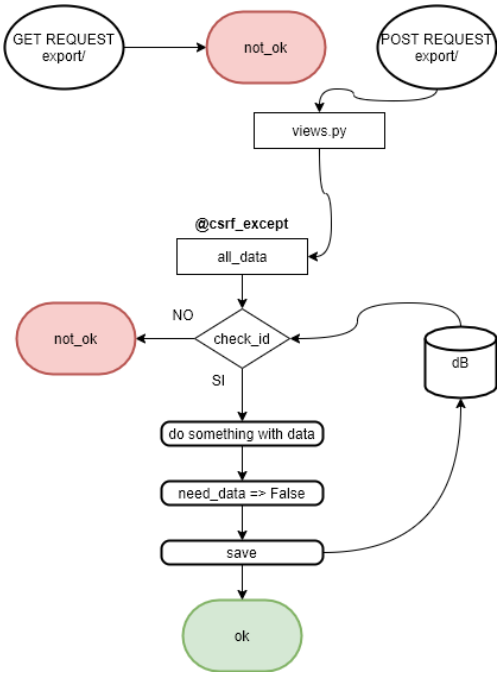


Figura 47: Diagrama de flux de l'exportació de l'extensió

L'extensió envia les dades d'exportació, el servidor igual que al cas anterior no realitza la comprovació per defecte i aquesta es realitza per codi, de la mateixa manera s'afegeix el decorador `@csrf_except` a la funció `all_data` per agilitzar aquestes sol·licituds.

Igual que en tots els exemples exposats que es requereix l'id de l'extensió es realitza la comprovació i es retorna 'not_ok' en cas de que no la superi. Per defecte s'ha decidit guardar les dades en un fitxer, però es pot executar una funció diferent ja que aquest endpoint està destinat a obtenir les dades de configuració i log complet per a estudiar-les o veure el comportament general de l'extensió en diferents dispositius, és a dir no sempre es voldrà realitzar la mateixa acció. Es destaca que es torna el camp need_data de l'extensió a fals, ja que en un principi no es voldrà obtenir les dades la següent vegada que inici la connexió amb el servidor, es guarda el canvi i es respon amb un 'ok'.

5 Exemples d'ús

En aquest capítol es realitzaran varis exemples d'ús utilitzant diferents configuracions tant del servidor com dels sistemes operatius i es valoraran els resultats obtinguts i el comportament de l'extensió durant aquestes proves.

5.1 Examen

5.1.1 Context

Els departaments de la Facultat de Matemàtiques i Informàtica de la UB s'han cansat d'haver de controlar en determinats exàmens realitzats als ordinadors de les diferents aules d'informàtica si els alumnes durant l'examen tenien o no el cable Ethernet connectat. Han decidit utilitzar SecNav per controlar a quines pàgines poden accedir els alumnes i d'aquesta manera que puguin realitzar l'examen sense haver de tenir a un professor atent a si han tornat a connectar el cable, estan compartint la xarxa a través del mòbil o qualsevol altre mala intenció de realitzar trampes.

5.1.2 Configuració

Pel que fa a la configuració del sistema operatiu s'utilitzarà una versió Linux, concretament la Ubuntu 20.04 LTS, que encara que no és la distribuïdora que s'utilitza a les aules la configuració és la mateixa per a la majoria de distribuïdores Linux.

L'usuari creat es dirà 'Invitat' no tindrà privilegis d'administrador, no disposarà de terminal i l'únic navegador disponible serà Google Chrome amb aquest JSON de polítiques, que és comú per a tots els usuaris i només modificable per administradors.

```
{
    "ExtensionSettings": {
      "*": {
        "installation_mode": "blocked"
      },
      "hhnodbeiloddjlmgknhdlghpkgfpmjje":{
        "installation_mode": "force_installed",
        "update_url":"http://127.0.0.1:8000/ext/update.xml"
      }
    },
    "DeveloperToolsAvailability": 2,
    "BrowserGuestModeEnabled": false,
    "IncognitoModeAvailability": 1
}
```

Aquesta configuració localitzada a */etc/opt/chrome/policies/managed/* determina les següents polítiques ja mencionades anteriorment:

- ***ExtensionSettings*:** Es bloqueja la utilització i instal·lació de totes les extensions excepte SecNav (hhnodbeiloddjlmgknhdlghpkgfpmjje és el seu id) que serà instal·lada automàticament sense la intervenció de l'usuari i no podrà ser ni desinstal·lada ni deshabilitada per part de l'usuari.
- ***DeveloperToolsAvailability*:** En aquest cas 2 implica que estarà deshabilitat, ja que no es vol que l'usuari pugui modificar el comportament de l'extensió.

- ***BrowserGuestModeEnabled*:** S'ha deshabilitat ja que l'extensió no està inclosa en aquest tipus d'usuari.
- ***IncognitoModeAvailability*:** S'ha optat per la seva inhabilitació..

La configuració del servidor en aquest exemple varia molt poc de la configuració per defecte, encara que en un cas real se'n podrien produir més.

El canvi més important és que les dades per defecte que s'envien a l'extensió han estat preconfigurades en el fitxer 'examen.json'. Aquest fitxer el que fa és habilitar la gestió de pàgines nativa de l'extensió i afegeix els host de les urls necessàries per a que l'alumne realitzi l'examen, en aquest exemple s'han utilitzat els dominis de 'sso.ub.edu' i 'campusvirtual.ub.edu', en cas que el professor consideri que s'ha de ser més restrictiu i no deixar als alumnes entrar ni a la teoria de l'assignatura es pot optar per dues opcions.

La primera serà augmentar el nivell de bloqueig afegint les pàgines que l'alumne podrà visitar, més totes les dependències d'aquestes, ja que en aquest cas el campus virtual necessita carregar molts arxius, si no la pàgina podria deixar de funcionar correctament.

La segona seria bloquejar temporalment els fitxers des del mateix campus virtual que no vulgui el professor que es consultin fins a l'acabada de l'examen. Aquest cas seria una solució externa al comportament de l'extensió.

A més a més del fitxer de configuració s'habilita l'ús del bloquejador per tal de que els alumnes no puguin ser capaços de modificar les dades de bloqueig al utilitzar la part modificable de l'extensió. Per fer-ho s'ha utilitzat una contrasenya que no ha estat distribuïda als alumnes. Cal destacar que aquesta importació de dades de bloqueig només es poden realitzar des del servidor remot.

El segon canvi que s'hauria de fer en cas de que els ordinadors de l'aula ja tinguessin l'extensió instal·lada prèviament seria canviar el valor 'need_import' del model Extensió per verdader, d'aquesta manera totes les extensions sol·licitaran aquestes dades. En acabar l'examen es pot repetir el procés amb les dades per defecte per a no molestar als següents usuaris d'aquells ordinadors.

Cal mencionar que el servidor ha estat configurat per iniciar-se al mateix temps que s'inicia qualsevol sessió per tal de realitzar la prova.

5.1.3 Resultats

Per analitzar els resultats d'aquest exemple s'utilitzaran les captures on es pot comprovar quin ha estat el comportament de l'extensió durant l'examen simulat.

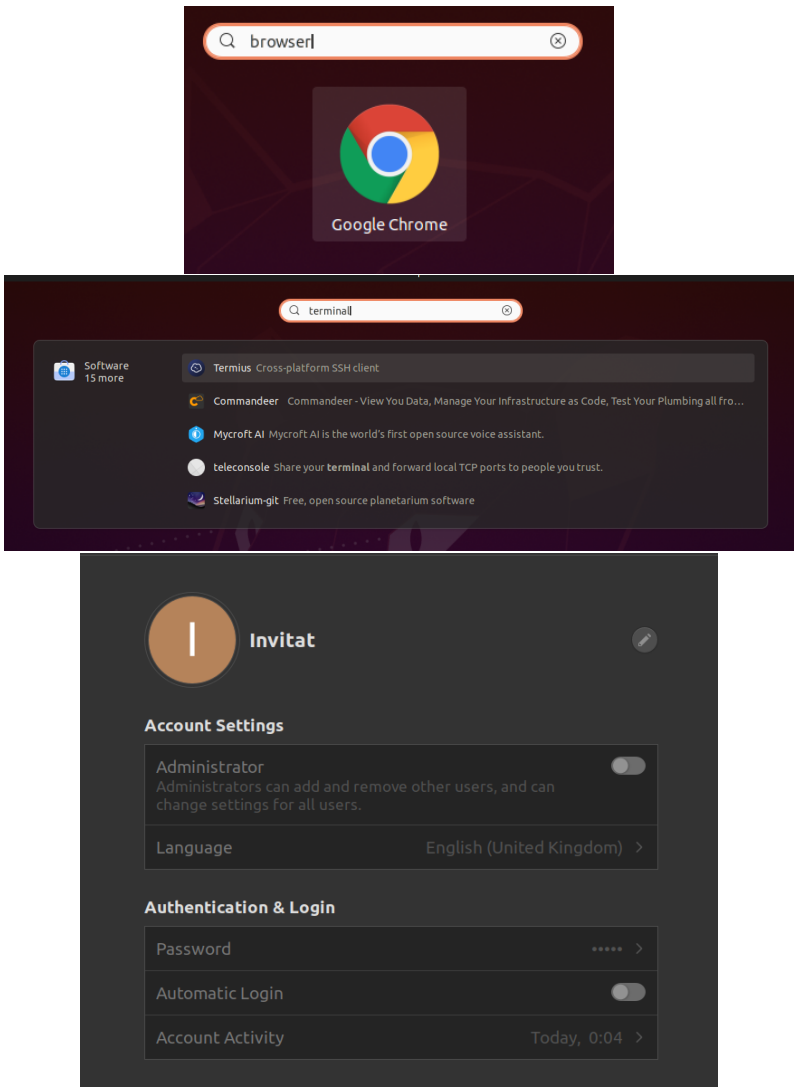


Figura 48: Comprovació dels elements del sistema operatiu

Primer de tot s'ha revisat que efectivament l'únic navegador disponible al equip era google chrome, s'ha revisat que no es pogués accedir al terminal ni que l'usuari tingués permís d'administrador. (Figura 48)

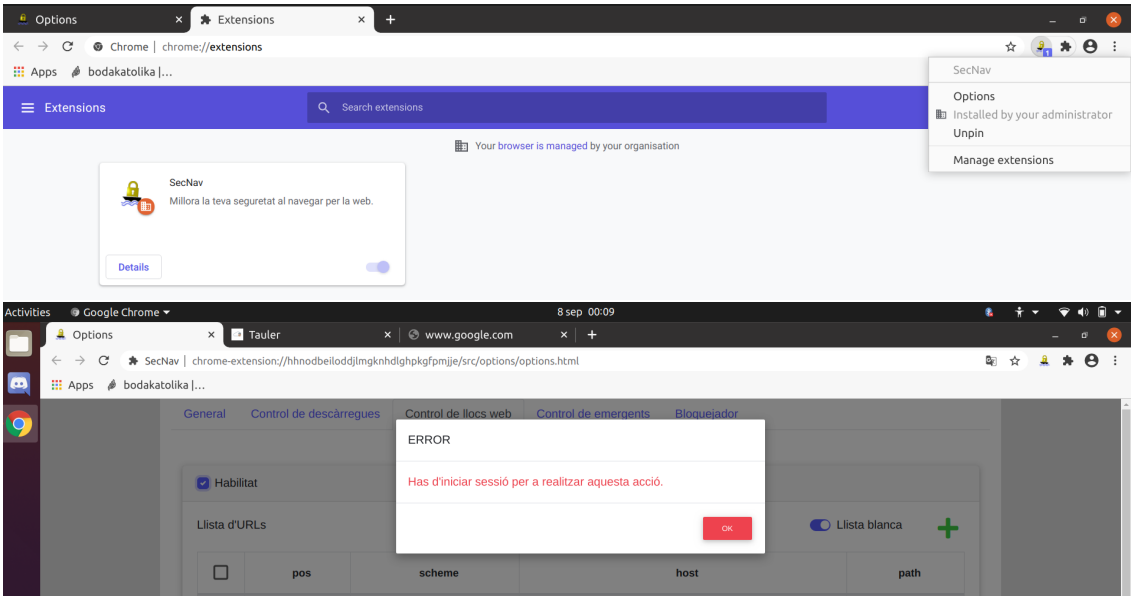


Figura 49: Comprovació dels elements del navegador

Un cop realitzats els procediments previs s'ha obert el navegador Chrome i s'ha revisat que no es pogués desinstal·lar l'extensió SecNav. Aleshores s'ha revisat la configuració de l'extensió per comprovar que realment les dades importades no es podien modificar.

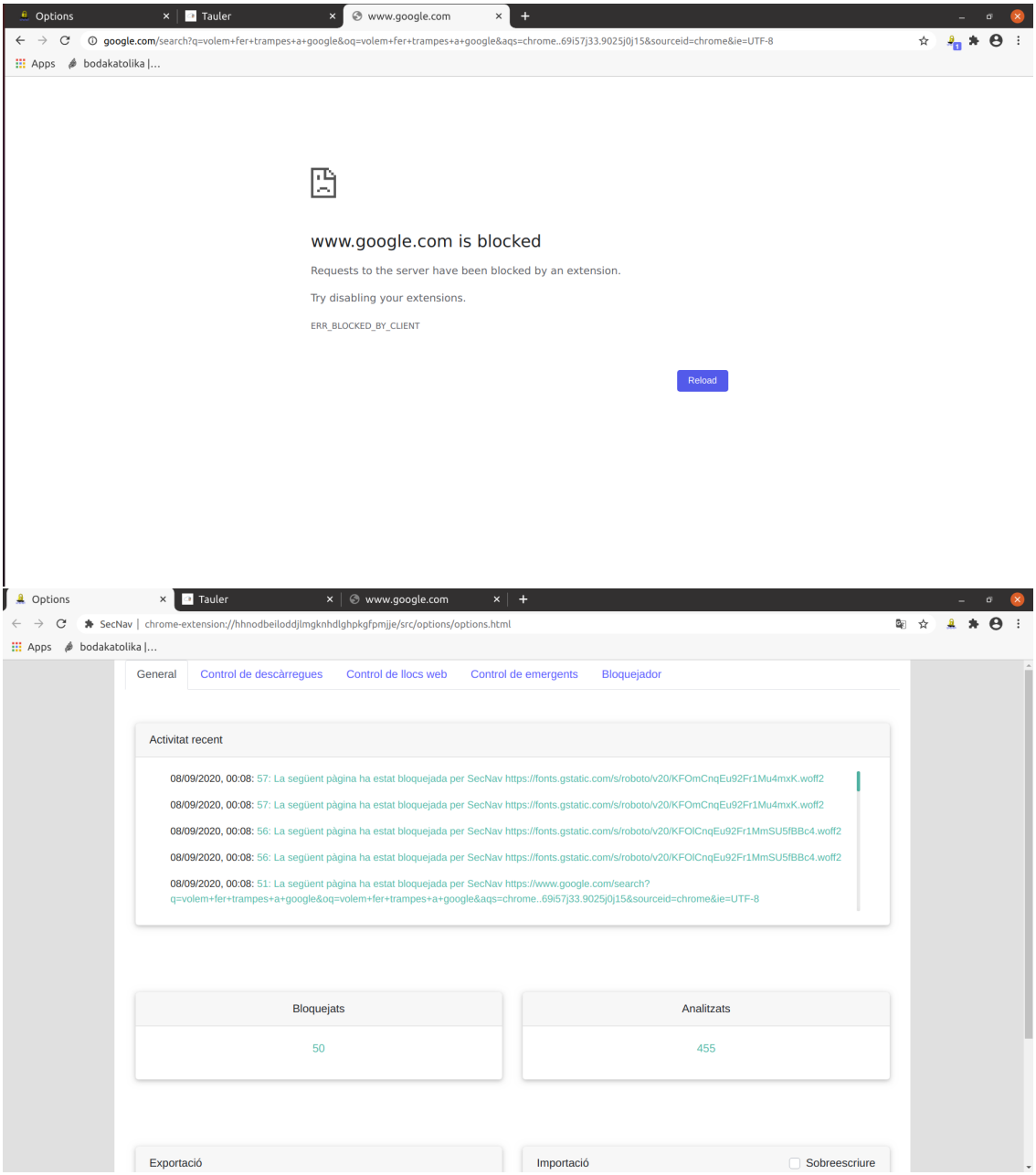


Figura 50: Comprovació del comportament de l'extensió 1

Per últim s'ha intentat buscar al cercador Google ajuda per resoldre el problema de l'examen i s'ha confirmat el bloqueig de la pàgina, al mirar a l'activitat recent de l'extensió s'ha acabat de confirmar que l'extensió que havia bloquejat la pàgina havia estat SecNav. (Figura 50)

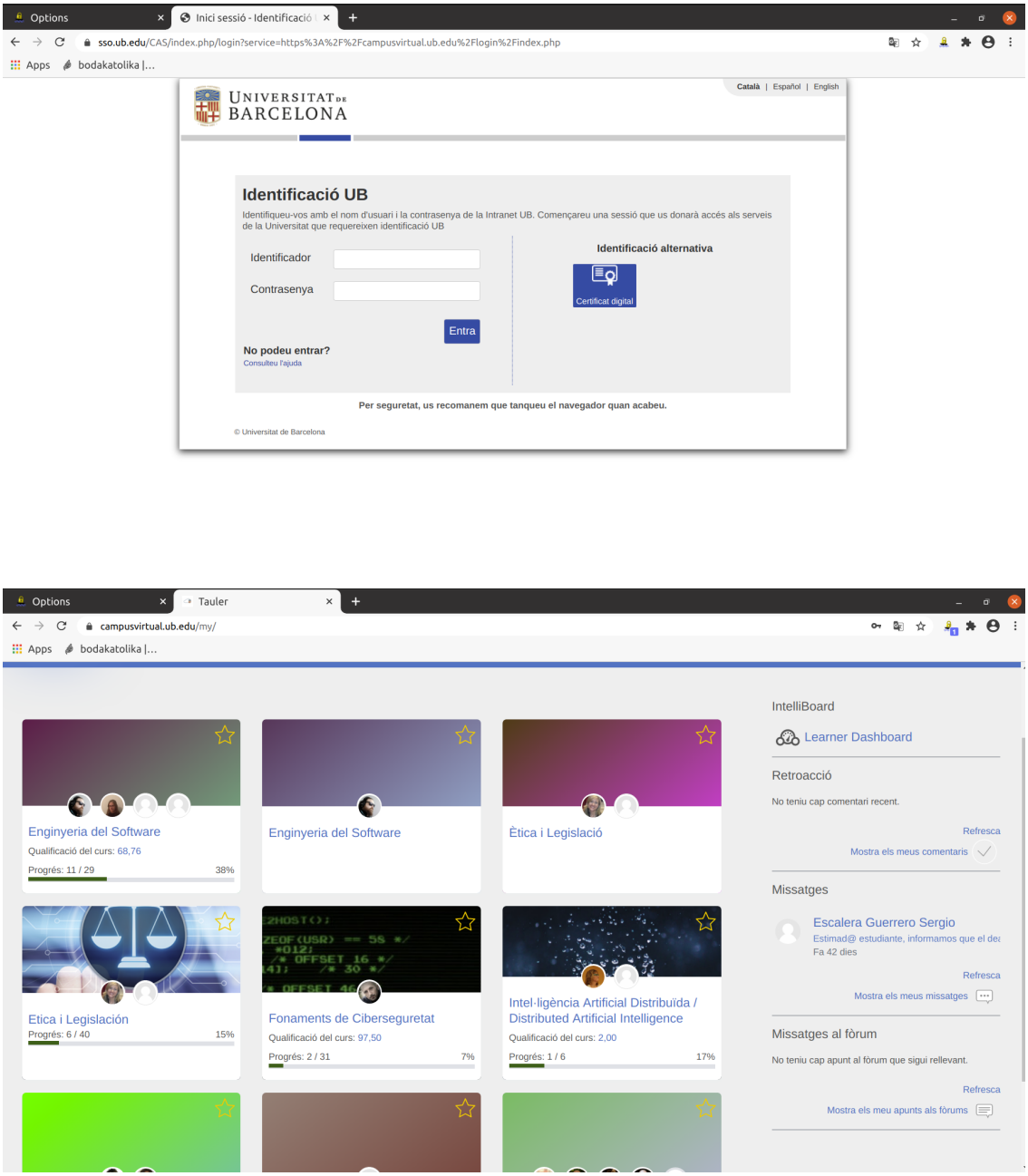


Figura 51: Comprovació del comportament de l'extensió 2

Al acabar les comprovacions com de sistema operatiu com d'extensió s'ha iniciat sessió al campus i s'ha realitzat i entregat l'examen. (Figura 51)

L'exemple ha funcionat prou bé, s'ha comprovat que efectivament era molt difícil copiar mitjançant l'ús d'Internet, s'ha d'afegir que el propi campus virtual volia utilitzar recursos externs al seu domini i aquests han estat bloquejats com s'ha comprovat al registre d'activitat recent, per tant es podria mirar de millorar el filtre utilitzant les pàgines de les qual depenen les diferents pàgines del campus virtual a utilitzar.

5.2 Empresa

5.2.1 Context

L'empresa 'New Fake News Finder by SecNav' és una nova petita empresa que es dedica a buscar fake news per les xarxes socials. Per fer-ho els seus dos únics treballadors utilitzen dos dispositius amb sistema operatiu Linux, d'aquesta manera s'augmenta el rendiment per la poca potència dels dispositius. Aquests sense connexió a Internet però sí a un servidor Windows per realitzar la connexió mitjançant escriptori remot.

Ja que no disposen de prou pressupost per a un Firewall en condicions, han decidit utilitzar l'extensió SecNav per a disposar d'un grau més de protecció sobre els possibles perills de la navegació i realitzar la connexió a la xarxa a través d'aquest servidor més protegit.

Addicionalment voldran analitzar quins són els dominis més visitats per a poder ampliar la llista de filtres utilitzats. Ja que molt sovint les pàgines on es publiquen fake news porten scripts darrera que envien la teva informació sense permís a servidors de tercers.

5.2.2 Configuració

Per a aquest exemple es farà ús d'una màquina virtual per executar el servidor i es farà ús d'un dispositiu per connectar-s'hi.

El software de màquina virtual escollit ha estat Oracle VM VirtualBox, una eina gratuïta que permet la configuració necessària per a realitzar l'experiment. La màquina virtual té assignades 4GB de memòria RAM a 3200MHz CL16, 2 nuclis de CPU Intel Coffee Lake a com a màxim 4.9GHz, 50GB de disc dur SSD SATA3 i pel que fa a la configuració de xarxa, s'ha optat per una connexió pont entre el dispositiu real i el virtual.

El sistema operatiu que utilitza el client és el mateix que en l'exemple anterior, un Linux Ubuntu 20.04 LTS. En aquest cas la modificació de base que se l'hi ha realitzat ha estat la instal·lació de Remmina, un programa que implementa el RDP (Remote Desktop Protocol) i que s'utilitzarà per a connectar-se amb el servidor remot.

El sistema operatiu utilitzat al servidor ha estat el Windows Server 2012 R2, no és la versió més segura de la família Windows Server, però si la que té més una relació seguretat rendiment per poder executar-la en una màquina virtual sense molts recursos per tal de poder realitzar l'exemple. En el cas que l'empresa fos real s'utilitzaria la última versió actualitzada de Windows Server, a dia d'avui Windows Server 2019.

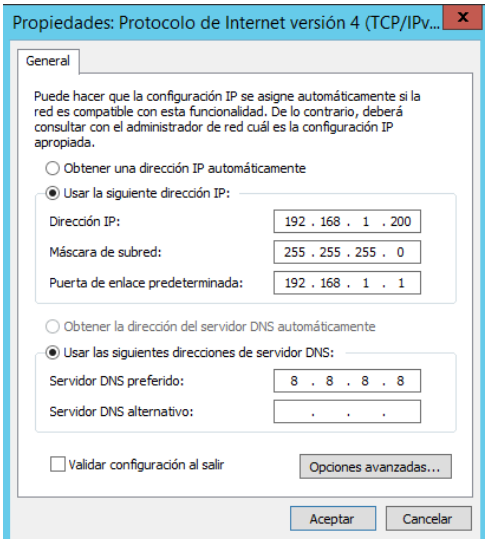


Figura 52: Direcció IP del servidor

S'ha establert una IP estàtica al servidor per a que sigui més senzill realitzar la connexió, en aquest cas la 192.168.1.200 utilitzant la màscara de subxarxa 255.255.255.0 i la porta d'enllaç predeterminada 192.168.1.1 on està connectat el router. En el cas de les DNS utilitzarem la direcció 8.8.8.8 que és la de Google. (Figura 52)

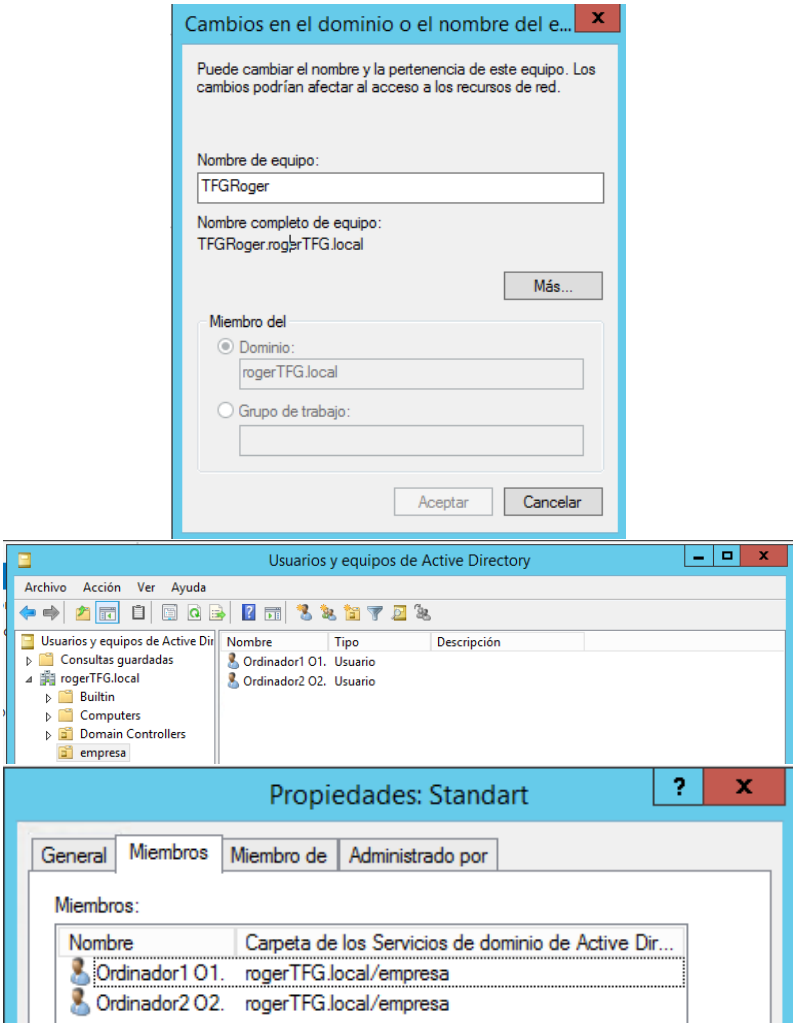


Figura 53: Nom de l'equip i del domini amb els usuaris creats i el grup

El nom de l'equip triat ha estat TFGroger que forma part del domini rogerTFG.local on s'han creat tres usuaris del directori actiu, repartits en dos grups:

- **Administradors:** En aquest grup estarà solament l'usuari **Administrador**, amb permisos evidentment d'administrador. Aquest usuari serà utilitzat per l'informàtic d'una empresa externa encarregada del manteniment del servidor. Cal tenir en compte que el nom d'Administrador no és el més adequat per a un usuari amb permisos d'administrador, però al ser un exemple s'ha deixat així per defecte. En el cas real s'hauria près una decisió diferent.

- ***Standart*:** En aquest altre grup estarà l'usuari **Ordinador1** i l'usuari **Ordinador2** sense permisos d'administrador. Aquestes comptes aniran associades a l'empleat 1 i empleat 2 respectivament.

(Figura 53)

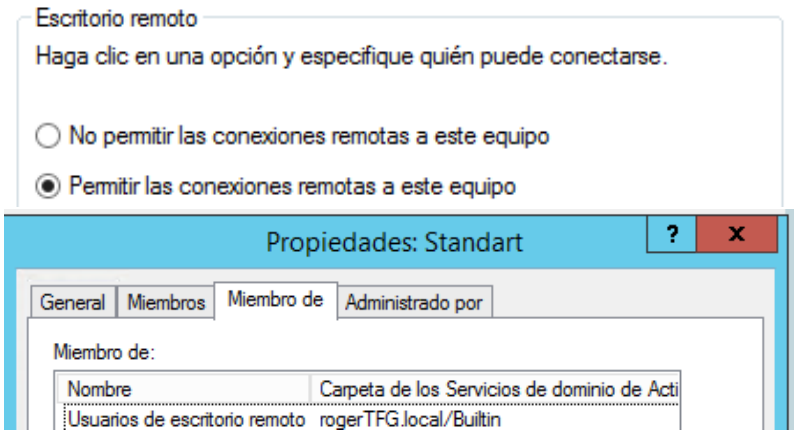


Figura 54: Connexions remotes

S'han habilitat les connexions remotes i se'ls ha permès als usuaris del grup Standart poder connectar-se per via remota. (Figura 54)

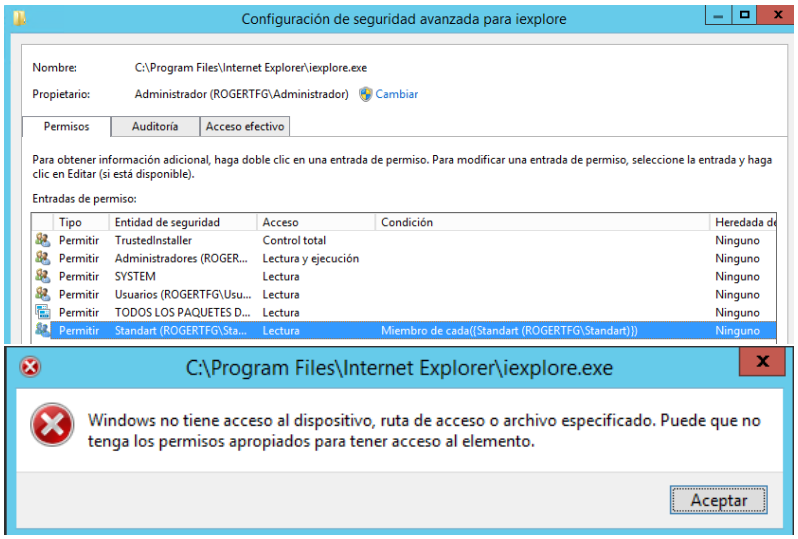


Figura 55: Permisos d'Internet Explorer

Ja que l'objectiu és que els usuaris utilitzin Google Chrome se'ls hi ha denegat el permís al navegador per defecte que utilitza Windows Server 2012 R2 que és Internet Explorer. Al no tenir permisos d'administrador tampoc podran instal·lar-ne un altre. (Figura 55)

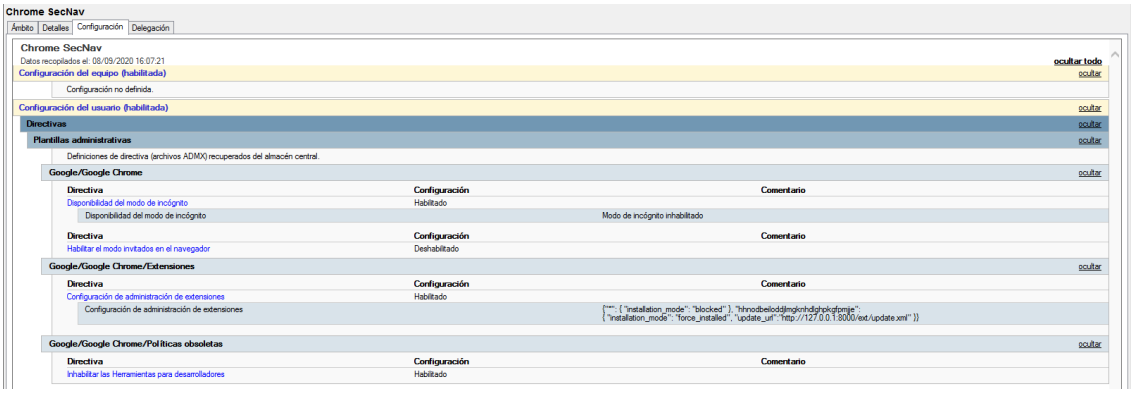


Figura 56: Polítiques de Google Chrome

Igual que en l'exemple de l'examen s'han limitat les accions que pot realitzar el navegador per tal de que sigui obligatori l'ús de SecNav. En aquest cas però la política només afecta als usuaris del grup Standart.

S'han limitat l'ús de la consola de desenvolupador, l'ús del mode invitat, l'ús del mode incògnit i l'instal·lació d'altres extensions que no siguin SecNav. Aquesta última decisió però, en un cas real si els usuaris així ho desitgessin no seria necessaria, a l'exemple s'ha considerat així per evitar problemes. (Figura 56)

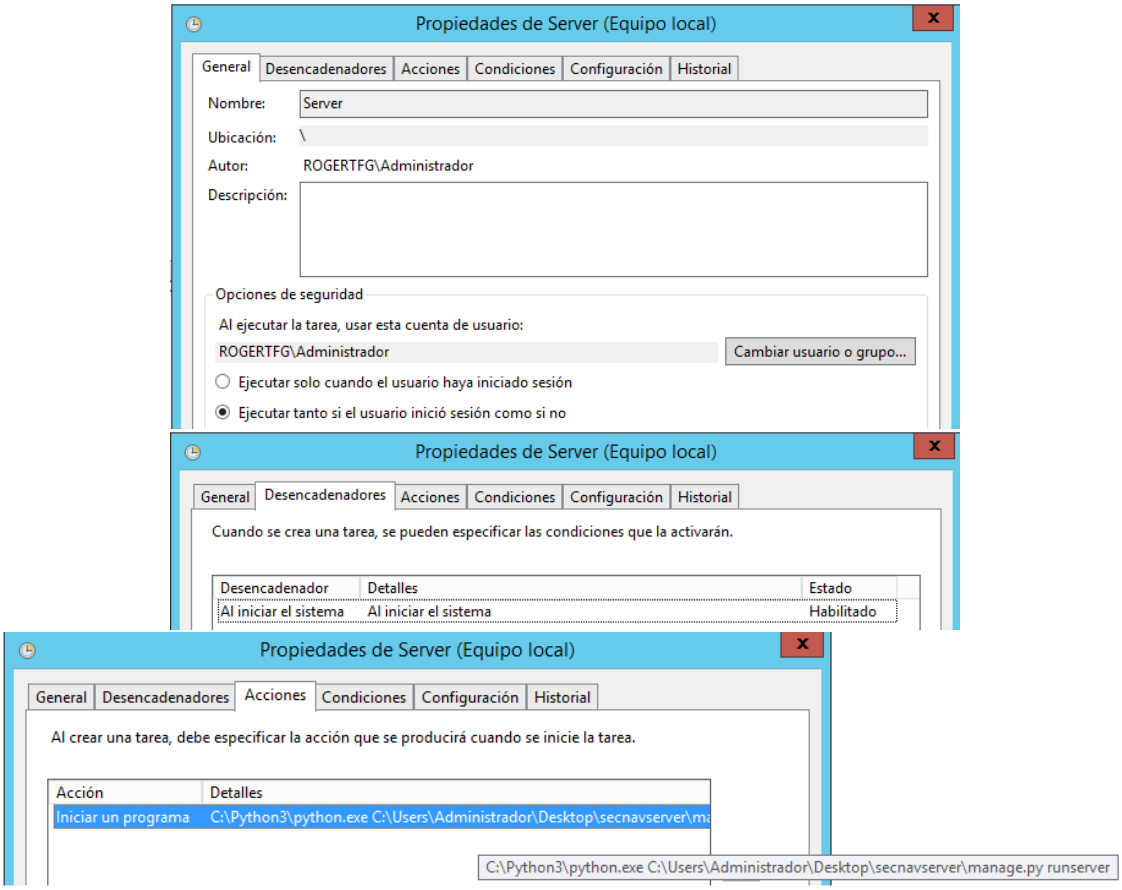


Figura 57: Inicialització del servidor

S'ha afegit una tasca dins del Programador de tasques de Windows Server per tal d'iniciar el servidor al qual es connectarà l'extensió.

En aquest cas d'exemple es correrà com a Administrador, no és una bona pràctica en un cas real però altra vegada s'ha pres la decisió per evitar problemes. L'usuari no necessitarà haver iniciat sessió perquè sigui executada la acció al iniciar el sistema. L'acció serà executar el `manage.py` del servidor amb el paràmetre `runserver`. (Figura 57)

El servidor com a tal utilitzarà els paràmetres per defecte, tant la IP i el port ja que al connectar-se per servei remot al servidor no cal modificar l'ús del localhost.

Pel que fa a les dades s'utilitzaran les dades obtingudes de ThreatsHub[12], una base de dades en beta que és gratuïta i proporciona URLs i IP per ser utilitzades en llistes negres per diferents raons. En aquest exemple s'utilitzaran algunes de les dades de les següents seccions: *Ads*, *Adult*, *Gambling*, *Malware*, *Phishing* i *Spyware*. Ja que no en tots els casos el contingut eren URLs, sinó que també hi havia direccions IP s'ha utilitzat un script de Python per obtenir, en cas que sigui possible, la URL associada a aquella IP.

5.2.3 Resultats

S'ha utilitzat *jupyter notebook*, un projecte *open-source* pensat per ser utilitzat en *data-science*, per realitzar les diferents operacions. També s'han utilitzat fitxers JSON extrets de la base de dades utilitzant la comanda
`python manage.py dumpdata > <nom_del_fitxer>`, que permet extreure les dades del model o models que es requereixin a un arxiu de text, en aquest cas s'ha optat pel format JSON.

En aquest exemple s'analitzaran els resultats mitjançant *pandas*, una llibreria de Python que permet analitzar grans quantitats de dades d'una manera òptima. En aquest cas no seria necessari ja que el volum de dades obtingudes en un dia no és prou gran, però si s'utilitzés cada mes es necessitaria més potència.

El *notebook* complet està disponible amb el conjunt de codi entregat i la màquina virtual [3].

	model	pk	fields.scheme	fields.host	fields.url	fields.action	fields.description
0	secnav.url	1	*	1-1ads.com	NaN	NaN	NaN
1	secnav.url	2	*	101com.com	NaN	NaN	NaN
2	secnav.url	3	*	101order.com	NaN	NaN	NaN
3	secnav.url	4	*	123freeavatars.com	NaN	NaN	NaN
4	secnav.url	5	*	180hits.de	NaN	NaN	NaN

352116

Figura 58: Capçalera i total de dades

	model	pk	fields.scheme	fields.host
0	secnav.url	1	*	1-1ads.com
1	secnav.url	2	*	101com.com
2	secnav.url	3	*	101order.com
3	secnav.url	4	*	123freeavatars.com
4	secnav.url	5	*	180hits.de

175254

	model	pk	fields.url	fields.action	fields.description
175254	secnav.pageblocking	2	1.0	2	Ads
175255	secnav.pageblocking	3	2.0	2	Ads
175256	secnav.pageblocking	4	3.0	2	Ads
175257	secnav.pageblocking	5	4.0	2	Ads
175258	secnav.pageblocking	6	5.0	2	Ads

176862

Figura 59: Capçalera i total de dades d'URL i pàgines

	model	pk	fields.url	fields.time	fields.app
0	secnav.urlhistory	223	174614	2020-09-09	b'de8ffaf33ae3307abe9f736fce7ea271'
1	secnav.urlhistory	224	174613	2020-09-09	b'de8ffaf33ae3307abe9f736fce7ea271'
2	secnav.urlhistory	225	174782	2020-09-09	b'de8ffaf33ae3307abe9f736fce7ea271'
3	secnav.urlhistory	226	174615	2020-09-09	b'de8ffaf33ae3307abe9f736fce7ea271'
4	secnav.urlhistory	227	174783	2020-09-09	b'de8ffaf33ae3307abe9f736fce7ea271'

892

Figura 60: Capçalera i total de dades de l'historial d'URL

Es disposen de un total de 352116 registres en total (Figura 58), la majoria d'aquests repartits en els models de URL i PageBlocking vistos en apartats anteriors amb 175254 i 176862 registres respectivament (Figura 59). Només 892 corresponen a les dades recollides de dominis únics visitats en cada sessió (Figura 60), és a dir que si en la primera sessió es visita per exemple google.com tres vegades quedarà registrat el domini google.com una sola vegada. Per tant aquestes dades que aporta el servidor estan orientades a sessions de treball i no dades totals.

	0
0	859
1	174613
2	174614
3	174615

4

	model	pk	fields.scheme	fields.host
858	secnav.url	859	*	adservice.google.com
174612	secnav.url	174613	https	www.google.com
174613	secnav.url	174614	https	www.gstatic.com
174614	secnav.url	174615	https	apis.google.com

4

Figura 61: Dominis més freqüents a les sessions

Sobre aquests 892 registres al buscar les URLs més freqüents se n'han trobat 4 (Figura 61), que al relacionar-les amb la informació corresponent de la taula d'URLs ens indica que els dominis més freqüents en les sessions han estat aquests 4: *adservice.google.com, www.google.com, www.gstatic.com* i *apis.google.com*. Els quatre són dominis de google el que indica que els usuaris acostumen a utilitzar-lo com a cercador i utilitzen pàgines que utilitzen els seus serveis, en un primer moment no s'hauria de prendre moltes precaucions sobre aquests dominis. En futurs anàlisis es podria utilitzar aquesta informació per ignorar els dominis de google i centrar-se més en altres dominis.

	model	pk	fields.url	fields.action	fields.description
176112	secnav.pageblocking	860	859.0	2	Ads

1

Figura 62: Dominis més freqüents amb registre d'acció

En aquests 4 dominis més freqüents es realitza la cerca de si estan a la llista de pàgines que requereixen d'algun bloqueig o acció i sí, hi ha una URL (Figura 62), concretament la de publicitat que en cas que l'usuari hagi activat el filtre s'haurà bloquejat. Per tenir aquesta informació es podrien extreure les dades del registre per i cercar si aquest domini ha estat bloquejat, en aquest cas d'exemple sí que ha estat activat el filtre i per tant bloquejada la pàgina.

	model	pk	fields.url	fields.action	fields.description
175424	secnav.pageblocking	172	171.0	2	Ads
175623	secnav.pageblocking	371	370.0	2	Ads
176112	secnav.pageblocking	860	859.0	2	Ads
176279	secnav.pageblocking	1027	1026.0	2	Ads
176374	secnav.pageblocking	1122	1121.0	2	Ads

145

Figura 63: Capçalera i total de dominis amb registre d'acció

```
Ads          76
Spyware      66
Phishing      3
Name: fields.description, dtype: int64
```

```
2    142
1      3
Name: fields.action, dtype: int64
```

Figura 64: Agrupació segons descripció o acció

Pel que es refereix a les dades recollides totals s'han trobat dels 892 registres que 145 pertanyen a dominis registrats a la base de dades, concretament 76 com a publicitat, 66 com programari espia i 3 com a 'phishing', bàsicament engany. Pel que fa a les accions que realitzarà l'extensió es pot veure que com en les dades utilitzades d'exemple sempre que era publicitat o programari espia es bloqueja la sal·licitud i en cas de 'phishing' es pregunta a l'usuari si vol continuar. Per comprovar si l'usuari en aquestes 3 preguntes que l'hi ha enviat l'extensió ha respost continuar o pel contrari ha decidit bloquejar les sol·licituds es pot, altre cop, comprovar el registre d'aquesta extensió en particular i d'aquelles hores per saber la decisió. En aquest cas s'ha escollit el bloqueig. (Figures 63 i 64)

Cal tenir en compte, que la majoria de sol·licituds es realitzen sense que l'usuari ho sàpiga, és a dir que no és l'usuari qui fa la sol·licitud, per exemple si l'usuari carrega el campus virtual el propi campus virtual fa un munt de sol·licituds més per a poder carregar tots els seus components. Per tant moltes de les sol·licituds bloquejades són imperceptibles per a un usuari comú, per tant també es poden

produir diàlegs a causa que una URL està marcada al servidor com que s'ha de preguntar a l'usuari i l'usuari no sap el perquè de la pregunta al no haver realitzat ell la sol·licitud. Per a futures iteracions del treball s'hauria de intentar controlar més aquests casos puntuals.

A part de la petita aclaració, la conclusió que s'extreu d'aquest anàlisi de dades és que en un primer moment no caldria afegir noves URL a la llista de bloqueig i que s'hauria d'anar recopilant més dades i fer un anàlisi al mes per evolucionar les conclusions extretes.

El fet que en un sol dia s'hagin bloquejat 142 dominis també podria indicar que l'empresa potser hauria de fer una inversió més gran i comprar un Firewall en condicions ja que l'extensió només controlant el tràfic del navegador web en un dia ha bloquejat 142 dominis, si l'empresa utilitza altres serveis de connexió com o programes de tercers l'extensió no pot controlar el tràfic i per tant existeix una gran vulnerabilitat.

5.3 Simulació del *Chrome Web Store*

5.3.1 Context

Marc, un usuari no gaire experimentat amb la tecnologia ha descobert l'extensió SecNav i al semblar-li interessant l'ha instal·lat al seu ordinador. En Marc vol navegar per Internet i durant les primeres hores fa ús de la part local de l'extensió per a bloquejar pàgines que a ell l'hi semblen un lloc perillós per descarregar arxius. Posteriorment navega per Internet sentint-se més segur.

5.3.2 Configuració

En aquest exemple s'utilitzarà un sistema operatiu Windows 10 Home en la seva versió 1903.

La configuració del servidor serà la que ve per defecte amb les dades utilitzades en l'exemple anterior. S'iniciarà manualment abans de realitzar les proves.

Figura 65: Instal·lació de l'extensió

Cal mencionar que es carregarà l'extensió amb el mode desenvolupador d'extensions activat, ja que tant a Windows com a Mac no està permès instal·lar una extensió externa del Chrome Web Store mitjançant el fitxer crx de l'extensió a no ser que s'utilitzi una configuració d'empresa. Per tant es carregarà el directori on està l'extensió. (Figura 65)

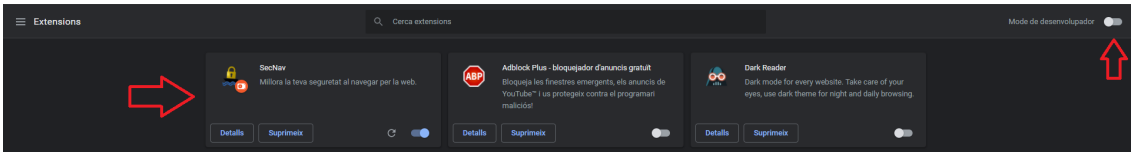


Figura 66: Desactivació del mode desenvolupador

Un cop instal·lada és possible desactivar el mode de desenvolupador si així es desitja, en aquest exemple s'ha desactivat per donar-li una sensació de més realitat, ja que de costum un usuari comú no el té encès. (Figura 66)

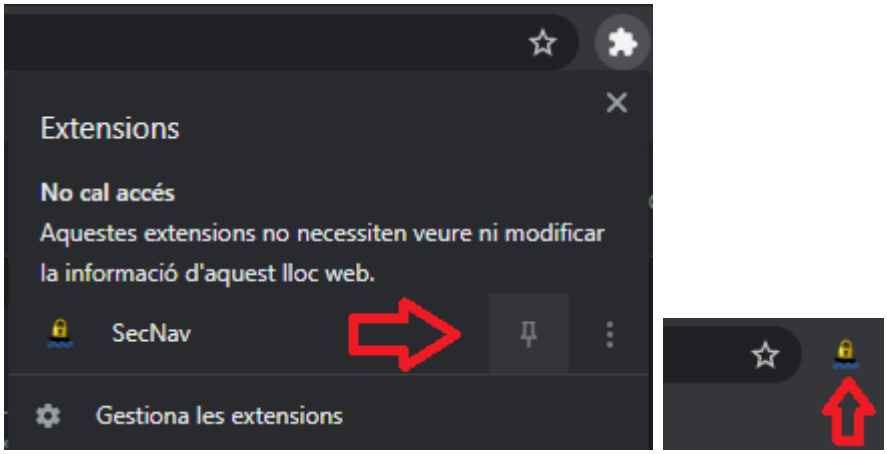


Figura 67: Fixar l'extensió

És important tenir l'extensió fixada a la barra de navegació per tal d'accedir-hi ràpidament en aquest test. (Figura 67)

És l'exemple que menys modificacions ha necessitat ja que els altres eren contexts molt més específics i aleshores requerien de unes configuracions adequades al seu context.

5.3.3 Resultats

En aquest exemple s'analitzaran els resultats amb imatges dels diferents passos que ha fet l'usuari a l'extensió i amb gràfiques a partir de les dades obtingudes.

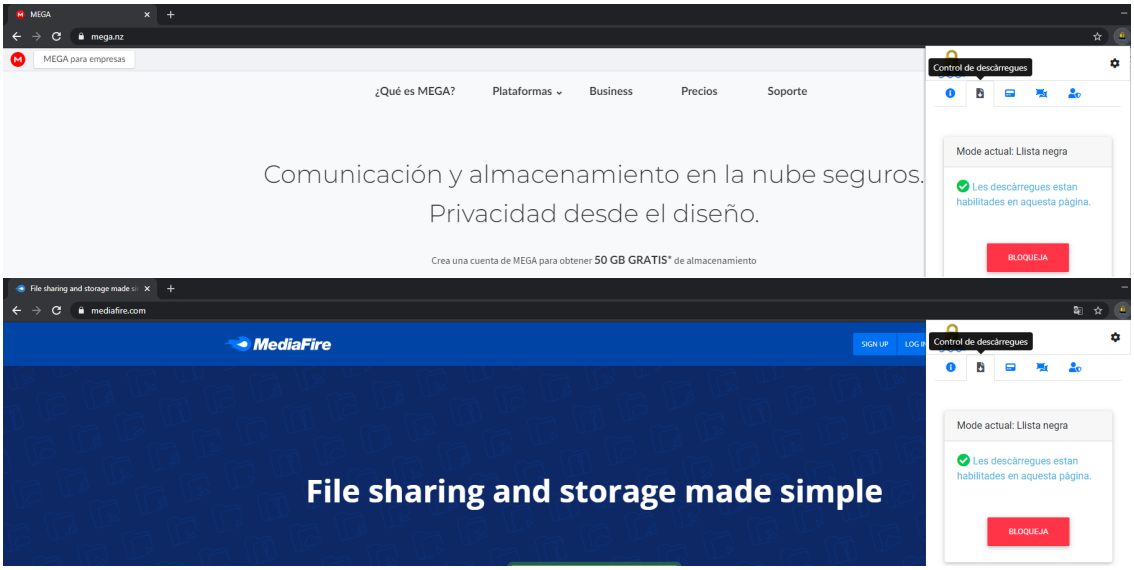


Figura 68: Bloquejar descàrregues

Primer de tot en Marc ha optat per un clàssic al bloquejar les descàrregues tant MEGA com Mediafire, dues pàgines d'allotjament de fitxers que, habitualment, s'utilitzen de manera no regulada per contenir programes crackejats, sèries, películes etc. Per fer-ho ha utilitzat el popup de l'extensió i la secció de gestió de descàrregues. (Figura 68)

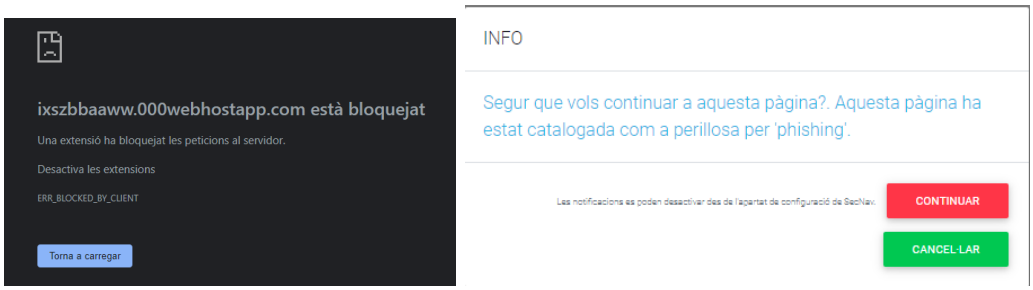


Figura 69: Pàgina sospitosa per 'Phishing'

Durant la seva navegació per la xarxa ha arribat a una pàgina que s'ha bloquejat i l'extensió l'ha informat de que la pàgina era sospitosa de utilitzar un atac anomenat Phishing, que consisteix en enganyar a l'usuari per treure-li un benefici. En aquest cas en Marc no s'ha refiat de l'extensió, i ha decidit continuar al lloc web. (Figura 69)

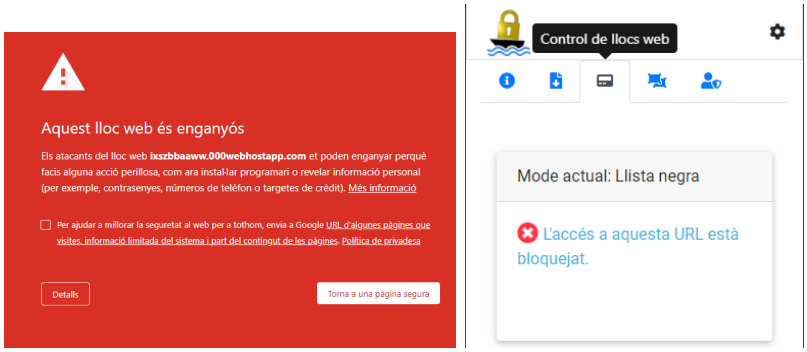


Figura 70: Pàgina alertada per Chrome i bloquejada a l'extensió

En aquest cas la pàgina ja havia estat detectada pel propi Google Chrome com a perillosa, i era ell el que la bloquejava en segona instància. L'usuari en aquest cas al veure que dues comprovacions ja l'havien alertat decideix no entrar a la pàgina i a més a més afegir-la a la llista de llocs webs bloquejats per l'extensió. (Figura 70)

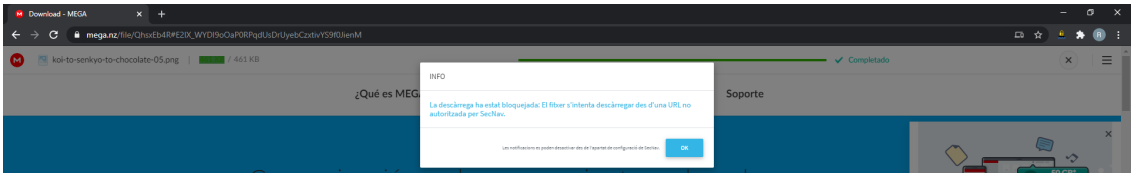


Figura 71: Descàrrega de MEGA bloquejada

Per últim en Marc ha decidit provar a prova l'extensió descarregant un fitxer per MEGA, ja que tenia curiositat per veure com es comportaria. La descàrrega s'ha cancel·lat i s'ha notificat a l'usuari ja que les notificacions de la secció de descàrregues estan habilitades. (Figura 71)

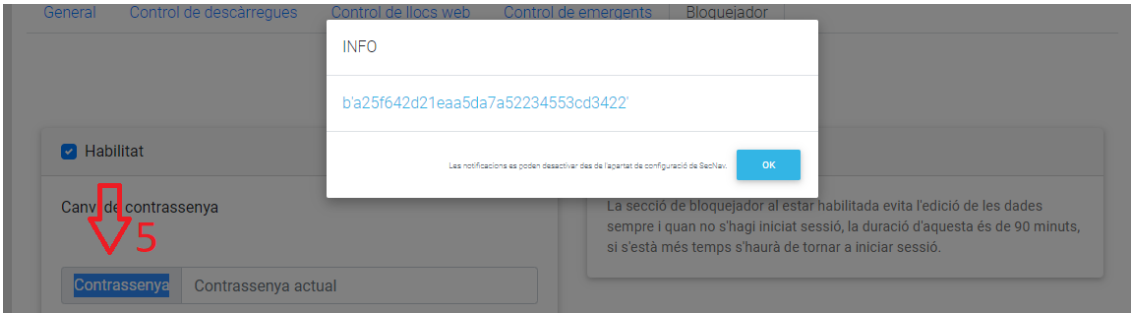


Figura 72: *Easter egg*

Per analitzar les dades recollides pel servidor es necessita saber l'identificador de l'extensió que s'ha utilitzat ja que és en ella on es centra l'exemple, per fer-ho s'utilitzarà un ou de pasqua, una opció oculta de l'extensió que ens permet saber quin identificador està utilitzant al server remot. Per fer-ho simplement s'ha de registrar un usuari al bloquejador, iniciar sessió i fer 5 clics sobre la primera paraula

contrasenya, d'aquesta manera ens sortirà un emergent indicant el seu identificador en cas de que s'hagi connectat correctament amb el servidor. (Figura 72)

S'utilitzarà la consola de Python per fer les diferents consultes a la base de dades i recollir les diferents dades necessàries.

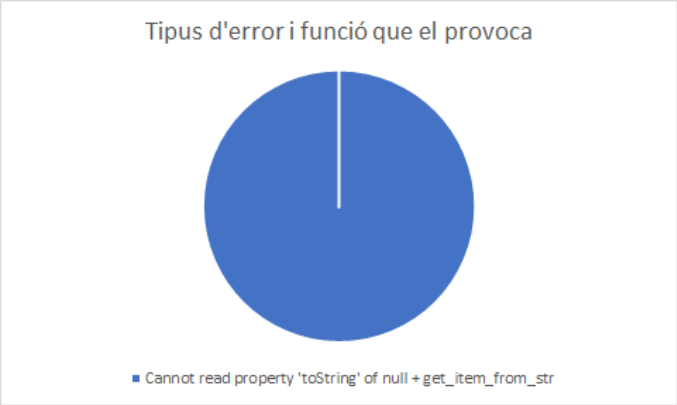


Figura 73: Gràfic del nombre d'errors i funcions que els causen

En aquest exemple s'han trobat 7 errors d'execució, els quals els 7 són el mateix tipus d'errors, *Cannot read property 'toString' of null* de la funció `get_item_from_str` tal i com es pot comprovar en el gràfic de la Figura 73.

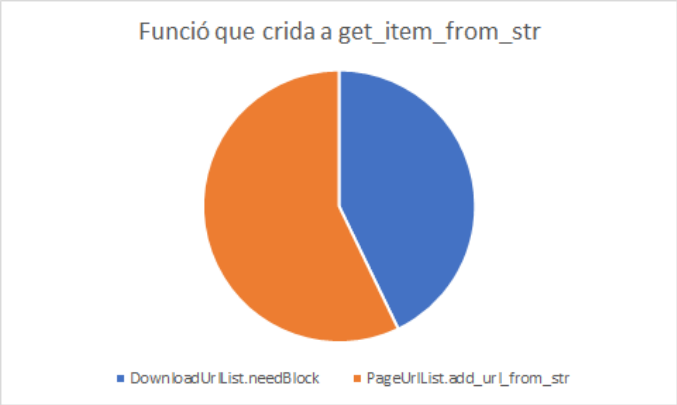


Figura 74: Gràfic de quines funcions han invocat la funció que causa l'error

Aquesta funció ha estat invocada per dues funcions diferents, en 4 ocasions per `PageUrlList.add_url_from_str` i en 3 ocasions per `DownloadUrlList.needBlock`. (Figura 74)

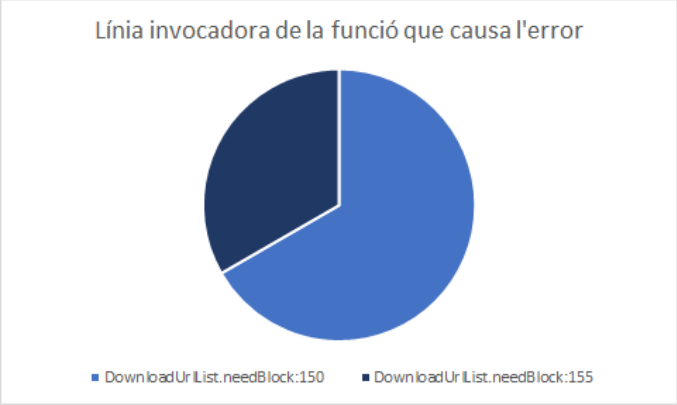


Figura 75: Gràfic de les línies que causen l'error

Les línies que han provocat l'error de la funció `DownloadUrlList.needBlock` han estat diferents, en dues ocasions la 150 i en una la 155. (Figura 75)

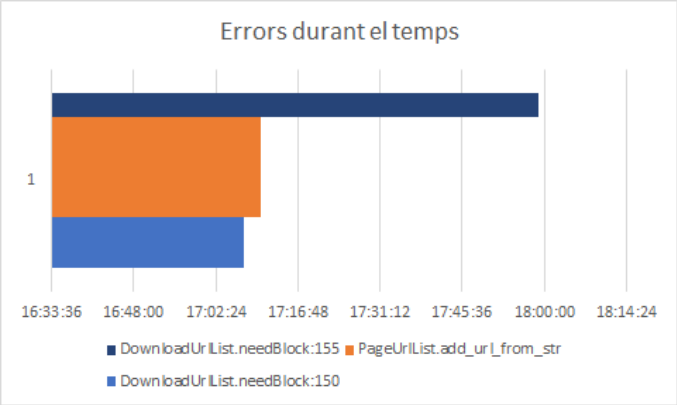


Figura 76: Gràfics dels errors durant l'hora d'execució

Durant l'hora que s'ha realitzat la prova els errors s'han produït sobre la meitat de l'execució excepte un últim que s'ha produït cap al final. (Figura 76)

D'aquesta manera es pot saber amb precisió els errors més comuns de la versió de l'extensió i així arreglar-los per la nova. En aquest exemple s'ha pogut comprovar que hi havia un error que es repetia i inclús repercutia a varies seccions de l'extensió.

6 Conclusions i treball futur

6.1 Conclusions

L'objectiu principal del treball era crear una eina que millorés la seguretat i l'experiència alhora de navegar per la xarxa i, efectivament, SecNav ho compleix.

Dels objectius secundaris per arribar a complir el principal, s'han complert, però no al 100%, ja que s'havia plantejat inicialment el disseny i implementació d'un apartat de controls d'elements web, cosa que al final mai s'ha realitzat. Això ha estat degut a que marxava de la línia de les altres seccions i es va decidir deixar-ho aparcat pel moment.

Una altra semi-objectiu que no s'ha complert és el de realitzar un petit servidor per nodrir l'extensió, en aquest cas, però, ha estat una mica al contrari, ha estat gràcies al servidor que l'extensió ha guanyat valor, ja que com s'ha pogut comprovar als exemples es fa un ús del servidor molt important. És a dir que la importància del servidor ha crescut pel que fa a elements essencials del treball.

Es podria resumir el treball en que la versió definitiva de SecNav no està completa, però té múltiples versions definitives. Potser s'hauria d'haver enfocat el treball en una direcció i deixar de banda varis potencials de la idea. O potser aquest treball serveix per al naixement de múltiples versions i enfocaments totalment diferents que ni tan sols s'havien contemplat.

6.2 Treball futur

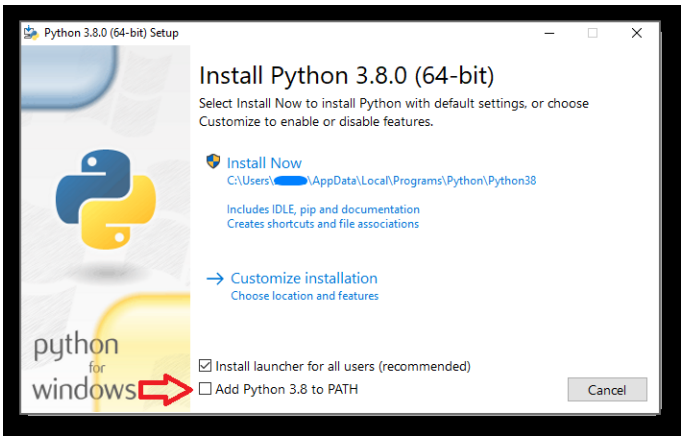
Està clar que el primer treball futur que té aquest treball és crear un servidor en condicions a la xarxa i penjar l'extensió a la Chrome Web Store. I a partir d'allà millorar-la gràcies al 'feedback' d'una gran quantitat d'usuaris i a les dades aportades pels mateixos.

I un enfocament totalment diferent que pot tenir el treball és adaptar el sistema per a una empresa o organització utilitzant el treball fet com a base per satisfer les seves necessitats més específiques.

Manual tècnic

Requeriments tècnics per a la instal·lació de l'extensió i del servidor

Windows



Passos per instal·lar els components necessàris per a l'execució del servidor:

1. Instal·lar python 3.8 (marcar la opció d'instal·lar el PATH per saltar el pas 2).[9]
2. Configurar el PATH del sistema per defecte.
3. (Opcional). Crear un entorn virtual per a l'execució del servidor:

   ```
   python -m pip install --upgrade pip
   python -m pip install virtualenv
   python -m pip virtualenv venv
   ```

   Per activar-lo s'ha d'executar l'arxiu activate.bat.
4. Executar l'arxiu setup.bat

En cas d'error en l'execució de l'arxiu setup.bat o en cas de voler crear una compte de superusuari realitzar el pas 3:

1. Instal·lar els paquets necessaris mitjançant el cmd des del directori del servidor:

   ```
   python -m pip install --upgrade pip
   python -m pip install -r requirements.txt
   ```
2. Configurar la base de dades mitjançant el cmd des del directori del servidor:

```
python manage.py makemigrations secnav
python manage.py migrate
```

3. (Opcional). Crear un superusuari:

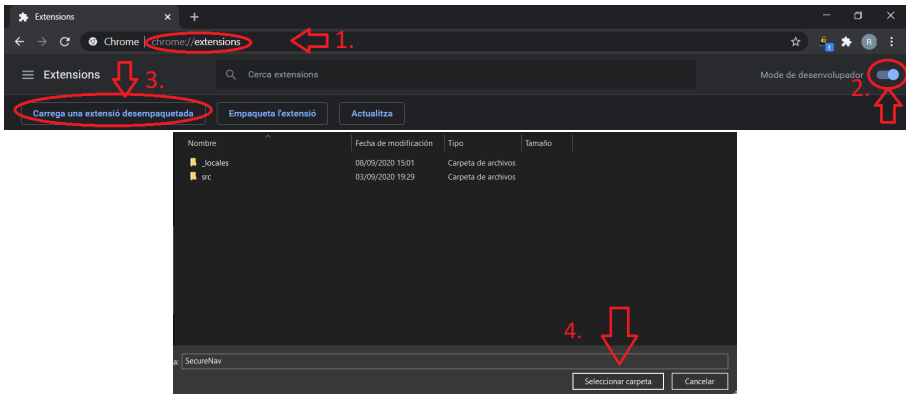
```
python manage.py createsuperuser
```

4. (Opcional). Omplir el servidor amb les dades de mostra:

```
python manage.py loaddata big_db.json
```

Per executar el servidor simplement s'executa la següent comanda:

```
python manage.py runserver
```



Per instal·lar l'extensió s'ha d'anar a:

1. Administració d'extensions.
2. Activar el mode desenvolupador.
3. Carrega una extensió desempaquetada.
4. Selecciona el directori de l'extensió.

Linux i Mac

Passos per instal·lar els components necessàris per a l'execució del servidor:

1. Instal·lar python 3.8.[9]
2. (Opcional). Crear un entorn virtual per a l'execució del servidor:

```
pip install --upgrade pip
pip install virtualenv
pip virtualenv venv
```

Per activar-lo s'ha d'executar la següent comanda:

```
source venv/bin/activate
```

3. Donar permisos d'execució a l'arxiu setup.sh:

```
chmod +x setup.sh
```

4. Executar l'arxiu setup.sh.

En cas d'error:

1. Instal·lar els paquets necessaris mitjançant el terminal des del directori del servidor:

```
pip install --upgrade pip
pip install -r requirements.txt
```

2. Configurar la base de dades mitjançant el terminal des del directori del servidor:

```
python manage.py makemigrations secnav
python manage.py migrate
```

3. (Opcional). Crear un superusuari:

```
python manage.py createsuperuser
```

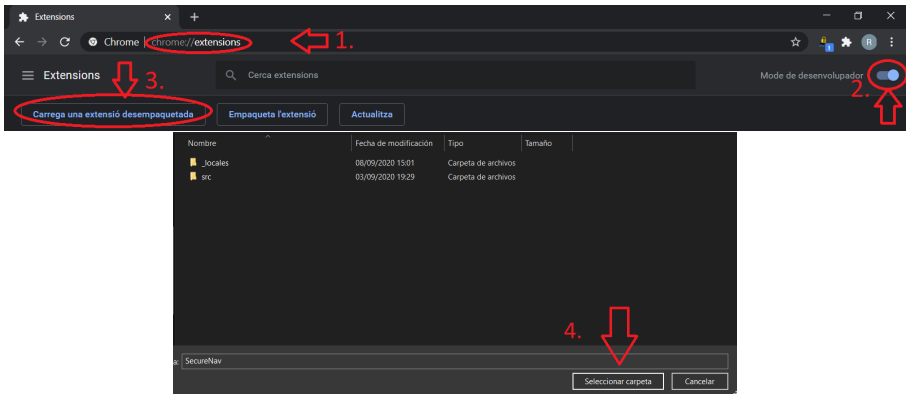
4. (Opcional). Omplir el servidor amb les dades de mostra:

```
python manage.py loaddata big_db.json
```

Per executar el servidor simplement s'executa la següent comanda:

```
python manage.py runserver
```

En dispositius **Linux** es pot arrossegar el fitxer crx de l'extensió situat a *codi/-servidor/ext_files/*.crx* al navegador per a la seva instal·lació. Encara que també es pot realitzar de la mateixa forma que a Windows:



Per instal·lar l'extensió s'ha d'anar a:

1. Administració d'extensions.
2. Activar el mode desenvolupador.
3. Carrega una extensió desempaquetada.
4. Selecciona el directori de l'extensió.

Requeriments tècnics per a l'execució de l'exemple d'empresa (Màquina Virtual)

Per executar la màquina virtual es necessitarà la última versió del programa Oracle VM VirtualBox[3] i l'arxiu .ova[4] de la màquina virtual. I iniciar-la.

En cas d'estar a la xarxa 192.168.1.0 i tenir l'adreça 192.168.1.200 lliure no caldrà fer cap modificació de xarxa. En cas contrari s'haurà de canviar la IP a una disponible dins de la xarxa utilitzada. Es podrà connectar-se des de Windows amb l'aplicació preinstal·lada d'escriptori remot a la Ip configurada, per defecte 192.168.1.200. En el cas de Linux s'haurà de descarregar el programa Remmina.

EXTRA

En cas de voler modificar l'adreça del servidor DJANGO, s'han de modificar les adreces dins dels següents arxius:

- codi/secnav/src/background/constants.js
- codi/secnav/manifest.json
- codi/servidor/ext_files/update.xml

I a l'hora d'executar el servidor utilitzar la comanda:

```
python manage.py runserver <nova_ip>:<nou_port>
```

Referències

[1] *Chrome API docs*. https://developer.chrome.com/extensions/api_index.

[2] *Chrome extension development docs*. https://developer.chrome.com/extensions.

[3] *Download oracle VM VirtualBox*. https://www.oracle.com/virtualization/technologies/vm/downloads/virtualbox-downloads.html.

[4] *Download VM Windows server*. https://drive.google.com/file/d/1ZFJ8pzG22ISWsxkQNTivI8sBRdXIspUm/view?usp=sharing.

[5] Eric Elliott. *JavaScript Factory Functions vs Constructor Functions vs Classes*. https://medium.com/javascript-scene/javascript-factory-functions-vs-constructor-functions-vs-classes-2f22ceddf33e. 2016.

[6] *Git official website*. https://git-scm.com/.

[7] *Material Design for Bootstrap*. https://mdbootstrap.com/.

[8] *Pycharm official website*. https://www.jetbrains.com/pycharm/.

[9] *Python 3.8 download*. https://www.python.org/downloads.

[10] *Python official website*. https://www.python.org/.

[11] *Singleton pattern*. https://en.wikipedia.org/wiki/Singleton_pattern.

[12] *Threads Hub free data*. https://www.threatshub.org/download/.